



**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code: 17332**

**Important Instructions to examiners:**

- 1) The answers should be examined by key words and not as word-to-word as given in the model answer scheme.
- 2) The model answer and the answer written by candidate may vary but the examiner may try to assess the understanding level of the candidate.
- 3) The language errors such as grammatical, spelling errors should not be given more Importance (Not applicable for subject English and Communication Skills).
- 4) While assessing figures, examiner may give credit for principal components indicated in the figure. The figures drawn by candidate and model answer may vary. The examiner may give credit for any equivalent figure drawn.
- 5) Credits may be given step wise for numerical problems. In some cases, the assumed constant values may vary and there may be some difference in the candidate's answers and model answer.
- 6) In case of some questions credit may be given by judgement on part of examiner of relevant answer based on candidate's understanding.
- 7) For programming language papers, credit may be given to any other program based on equivalent concept.

Q. No	Sub Q.N.	Answer	Marking Scheme
1.	(A) (a) Ans.	<b>Attempt any SIX of the following:</b> <b>Define Database. List any two properties of database.</b> <b>Definition:</b> Database is defined as collection of related data.  <b>Properties of database are:</b> i) Self-describing nature of a database system ii) Insulation between programs and data iii) Support of multiple views of the data iv) Sharing of data and multi-user transaction processing v) Controlling Redundancy.	12 2M  <i>Definiton 1M</i>  <i>Any 2 Properties 1/2 M each</i>
	(b) Ans.	<b>Define super key.</b> <b>Definition:</b> Super key of an entity set is a set of one or more attributes whose values uniquely determine each entity.	2M <i>Definiton 2M</i>
	(c) Ans.	<b>List the notation used in relational algebra.</b> 1. Selection ( $\sigma$ ) 2. Projection ( $\pi$ ) 3. Cartesian Product (X)	2M  <i>Any two notation</i>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION  
(Autonomous)  
(ISO/IEC - 27001 - 2005 Certified)

MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code:

17332

		4. Natural Join( $\bowtie$ ) 5. Union(U) 6. Set Difference ( - ) 7. Rename ( $\rho$ )	1 M each
(d) Ans.	<b>Define Group by clause.</b> The <b>SQL GROUP BY</b> clause is used along with the SQL aggregate functions and specifies the groups where selected rows are placed. When one or more aggregate functions are presented in the SQL SELECT column list, the SQL GROUP BY clause calculates a summary value for each group.		2M <i>Description 2M</i>
(e) Ans.	<b>What is SQL? State its features.</b> SQL is an acronym for Structured Query Language and is a standard relational query language used for creating and manipulating databases. <b>Features of SQL are:</b> <ul style="list-style-type: none"><li>• SQL allows the user to create, update, delete, and retrieve data from a database.</li><li>• SQL is very simple and easy to learn.</li><li>• SQL works with database programs like DB2, Oracle, MS Access, Sybase, MS SQL Server etc.</li></ul>		2M <i>Description 1M</i> <i>Any two features : 1/2 M each</i>
(f) Ans.	<b>Define index.</b> <b>Index:</b> An index is a schema object that can speed up the retrieval of rows by using pointer. An Index provides direct and fast access to rows in a table.		2M <i>Definition 2M</i>
(g) Ans.	<b>Define cursor.</b> The Oracle Engine uses a work area for its internal processing in order to execute an SQL statement. This work area is private to SQL's operations and is called a Cursor. <b>OR</b> A cursor is a temporary work area created in the system memory when a SQL statement is executed.		2M <i>Definition 2M</i>
(h) Ans.	<b>List advantages of Normalization.</b> <b>Advantages of the normalization.</b> <ul style="list-style-type: none"><li>• More efficient data structure.</li><li>• Avoid redundant fields or columns.</li><li>• More flexible data structure i.e. we should be able to add new rows and data values easily</li><li>• Better understanding of data.</li></ul>		2M <i>Any two advantages 1M each</i>



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code:

17332

		<ul style="list-style-type: none"><li>• Ensures that distinct tables exist when necessary.</li></ul>	
1.	(B) (a) Ans.	<p><b>Attempt any TWO of the following:</b></p> <p><b>Describe the advantages of Distributed Database Management System.</b></p> <p>Following are the advantages of distributed databases over centralized databases.</p> <p><b>1.Modular Development</b> – If the system needs to be expanded to new locations or new units, in centralized database systems, the action requires substantial efforts and disruption in the existing functioning.</p> <p><b>2.More Reliable</b> – In case of database failures, the total system of centralized databases comes to a halt. However, in distributed systems, when a component fails, the functioning of the system continues may be at a reduced performance. Hence DDBMS is more reliable.</p> <p><b>3.Better Response</b> – If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response.</p> <p><b>4.Lower Communication Cost</b> – In distributed database systems, if data is located locally where it is mostly used, then the communication costs for data manipulation can be minimized.</p> <p><b>5.Local autonomy:</b> Since data is distributed, a group of users that commonly share such data can have it placed at the site where they work, and thus have local control.</p> <p><b>6. Expandability:</b> Expansion can be easily achieved by adding processing and storage power to the existing network.</p>	08 4M  <i>Any 4 advantages : 1M each</i>
	(b) Ans.	<p><b>What is OUTER JOIN? Explain in detail.</b></p> <p>The SQL OUTER JOIN returns all rows from both the participating tables which satisfy the join condition along with rows which do not satisfy the join condition. The SQL OUTER JOIN operator (+) is used only on one side of the join condition only.</p> <p><b>Outer join are of three types:</b></p> <p><b>1. Left outer join</b> (also known as left join): this join returns all the rows from left table combine with the matching rows of the right</p>	4M  <i>Outer Join 2M</i>



**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code:**

<b>17332</b>
--------------

		<p>table. If we get no matching in the right table it returns NULL values.</p> <p><b>Left Outer Join syntax :</b> SELECT column-name-list from table-name LEFT OUTER JOIN table-name2 on table-name1.column-name = table-name2.column-name;</p> <p><b>2. Right outer join</b> (also known as right join): this join returns all the rows from right table are combined with the matching rows of left table .If we get no column matching in the left table, it returns null value.</p> <p><b>Right Outer Join Syntax:</b>SELECT column-name-list from table-name1 RIGHT OUTER JOIN table-name2 on table-name1.column-name = table-name2.column-name;</p> <p><b>3. The full outer join</b> keyword returns all records when there is a match in either left (table1) or right (table2) table records.</p> <p><b>FULL OUTER JOIN Syntax:</b>SELECT column_name(s)from table1 full outer join table2 on table1.column_name = table2.column_name;</p>	<p><i>Explanation of any 1</i> <b>2M</b></p>
(c) Ans.		<p><b>Describe Exception handling. Explain with example.</b> <i>(Note: Any example can be considered)</i></p> <p><b>Exception Handling:</b> Exception is nothing but an error. Exception can be raise when DBMS encounters errors or it can be raised explicitly. When the system throws a warning or has an error it can lead to an exception. Such exception needs to be handled and can be defined internally or user defined. Exception handling is nothing but a code block in memory that will attempt to resolve current error condition.</p> <p><b>Syntax:</b> <b>DECLARE</b>                   ; Declaration section                   ...executable statement;</p> <p><b>EXCEPTION</b> <b>WHEN ex_name1 THEN</b>                   :Error handling statements/user defined action to be carried out;</p>	<p><b>4M</b></p> <p><i>Description (syntax optional)</i> <b>2M</b></p>



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code:

17332

		<p><b>END;</b> <b>DECLARE</b>     s_rollNostudents.rollNo%type := 10;     s_namestudents.name%type;     s_addressstudents.address%type;</p> <p><b>BEGIN</b> SELECT rollNo, name, address FROM students WHERE rollNo = s_rollNo; dbms_output.put_line(s_rollNo    ' '    s_name    ' '    s_address);</p> <p><b>EXCEPTION</b> WHEN no_data_found THEN     dbms_output.put_line('No such student!'); WHEN others THEN     dbms_output.put_line('Error!');</p> <p><b>END;</b></p>	<p><i>Example 2M</i></p>
2.	(a) Ans.	<p><b>Attempt any FOUR of the following:</b> <b>Explain referential integrity constraints with example.</b> (Note: Any other example can be considered)</p> <p><b>Referential integrity constraint:</b></p> <ul style="list-style-type: none"><li>• It is used to established a parent child relationship between two tables.</li><li>• A value of foreign key is derived from the primary key.</li><li>• Primary key is defined in a parent table and foreign key is defined in child table. The child table contains the values for foreign key column which are present in parent tables primary key column but not other than that.</li></ul> <p><b>Syntax:</b> Create table tablename (column datatype size references parenttablename (primary key attribute)....)</p> <p><b>Example:</b> Create table product (EmpId number (4) references Emp (EmpId), PName varchar2(10));</p> <p><b>After table creation the foreign key is added as:</b> Alter table product add constraint fk_prod foreign key (EmpId) references Emp (EmpId);</p>	<p>16 4M</p> <p><i>Description 2M</i></p> <p><i>Example 2M</i></p>



**MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION**  
**(Autonomous)**  
**(ISO/IEC - 27001 - 2005 Certified)**

**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code:**

<b>17332</b>
--------------

<b>(b)</b> Ans.	<p><b>Explain single value and multivalued attribute of E-R model.</b></p> <p><b>i) Single Valued attribute:</b> Attributes that can have single value at a particular instance of time are called single valued.  <i>Example:</i> age of a person is a single-valued attribute.</p> <p><b>ii) Multi valued attributes:</b> A multi-valued attribute can have more than one value at one time.  <i>Example:</i> phone_no of a person is a multi-valued attribute since a person can have more than one phone_no.</p>	<b>4M</b>  <i>Description of each 2M</i>																																		
<b>(c)</b> Ans.	<p><b>Explain string, data and time functions of SQL.</b></p> <p>SQL string functions are used primarily for string manipulation. Following are the string functions used in SQL.</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <thead> <tr> <th style="text-align: center;">Function Name</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Initcap(char)</td> <td>Returns the input string with initial letter capitalized and all other character in lower case.</td> </tr> <tr> <td>Lower(char)</td> <td>Returns the input string with all letters in lower case.</td> </tr> <tr> <td>Upper(char)</td> <td>Returns the input string with all letters in Upper case.</td> </tr> <tr> <td>Ltrim(char,set)</td> <td>It removes or trims from the left of character string.</td> </tr> <tr> <td>Rtrim(char,set)</td> <td>It removes or trims specified set from the right of character string.</td> </tr> <tr> <td>Length(char)</td> <td>It returns the length of character string.</td> </tr> <tr> <td>Chr(n)</td> <td>Returns a character binary equivalent of n</td> </tr> <tr> <td>Ascii(char)</td> <td>Returns a decimal representation of a character</td> </tr> <tr> <td>Substr(char,m,n)</td> <td>It returns a portion of char, beginning at a character m, n character long, m can be +ve or -ve, n cannot be less than 1.</td> </tr> <tr> <td>Instr(char1, char2,[n [m]])</td> <td>Searches char 1 beginning with its nth character for the mth occurrence of char2 and return the position of character .i.e. the first character of this occurrence.</td> </tr> <tr> <td>Replace(char, searchstring, [repstring])</td> <td>It returns character string with each occurrence of searchstring replaced with [repstring].</td> </tr> <tr> <td>Translate(char, from, to)</td> <td>It returns char with all occurrences of each character in from replaced by its corresponding character in to. Character in char that are not in from are not replaced.</td> </tr> <tr> <td>Lpad(char1,length,char2)</td> <td>It returns char 1 left-padded to given length with the sequence of characters in char 2.</td> </tr> <tr> <td>Rpad (char1,length,char2)</td> <td>It returns char 1 right-padded to given length with the sequence of characters in char 2.</td> </tr> <tr> <td>Concat(char1, char2)</td> <td>Char 1 is concatenated to char 2 It merges two or more string value together. It is similar to concatenation operator   </td> </tr> <tr> <td>Soundex(char)</td> <td>Returns a character string containing the phonetic representation of char. This allows to compare words that are spelled differently but sound alike in English</td> </tr> </tbody> </table>	Function Name	Description	Initcap(char)	Returns the input string with initial letter capitalized and all other character in lower case.	Lower(char)	Returns the input string with all letters in lower case.	Upper(char)	Returns the input string with all letters in Upper case.	Ltrim(char,set)	It removes or trims from the left of character string.	Rtrim(char,set)	It removes or trims specified set from the right of character string.	Length(char)	It returns the length of character string.	Chr(n)	Returns a character binary equivalent of n	Ascii(char)	Returns a decimal representation of a character	Substr(char,m,n)	It returns a portion of char, beginning at a character m, n character long, m can be +ve or -ve, n cannot be less than 1.	Instr(char1, char2,[n [m]])	Searches char 1 beginning with its nth character for the mth occurrence of char2 and return the position of character .i.e. the first character of this occurrence.	Replace(char, searchstring, [repstring])	It returns character string with each occurrence of searchstring replaced with [repstring].	Translate(char, from, to)	It returns char with all occurrences of each character in from replaced by its corresponding character in to. Character in char that are not in from are not replaced.	Lpad(char1,length,char2)	It returns char 1 left-padded to given length with the sequence of characters in char 2.	Rpad (char1,length,char2)	It returns char 1 right-padded to given length with the sequence of characters in char 2.	Concat(char1, char2)	Char 1 is concatenated to char 2 It merges two or more string value together. It is similar to concatenation operator	Soundex(char)	Returns a character string containing the phonetic representation of char. This allows to compare words that are spelled differently but sound alike in English	<b>4M</b>  <i>Any 2 String functions 1M each</i>
Function Name	Description																																			
Initcap(char)	Returns the input string with initial letter capitalized and all other character in lower case.																																			
Lower(char)	Returns the input string with all letters in lower case.																																			
Upper(char)	Returns the input string with all letters in Upper case.																																			
Ltrim(char,set)	It removes or trims from the left of character string.																																			
Rtrim(char,set)	It removes or trims specified set from the right of character string.																																			
Length(char)	It returns the length of character string.																																			
Chr(n)	Returns a character binary equivalent of n																																			
Ascii(char)	Returns a decimal representation of a character																																			
Substr(char,m,n)	It returns a portion of char, beginning at a character m, n character long, m can be +ve or -ve, n cannot be less than 1.																																			
Instr(char1, char2,[n [m]])	Searches char 1 beginning with its nth character for the mth occurrence of char2 and return the position of character .i.e. the first character of this occurrence.																																			
Replace(char, searchstring, [repstring])	It returns character string with each occurrence of searchstring replaced with [repstring].																																			
Translate(char, from, to)	It returns char with all occurrences of each character in from replaced by its corresponding character in to. Character in char that are not in from are not replaced.																																			
Lpad(char1,length,char2)	It returns char 1 left-padded to given length with the sequence of characters in char 2.																																			
Rpad (char1,length,char2)	It returns char 1 right-padded to given length with the sequence of characters in char 2.																																			
Concat(char1, char2)	Char 1 is concatenated to char 2 It merges two or more string value together. It is similar to concatenation operator																																			
Soundex(char)	Returns a character string containing the phonetic representation of char. This allows to compare words that are spelled differently but sound alike in English																																			



**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code:**

<b>17332</b>
--------------

		<p><b>Date Manipulation Functions:</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">Sr.No.</th> <th style="width: 40%;">Format</th> <th style="width: 50%;">Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1.</td> <td>Month_between(d1,d2)where d1 and d2 are dates.</td> <td>Months_between finds the number of months between d1 and d2. If date d1 is later than d2 the result is positive. If date d1 is earlier than d2 the result is negative.</td> </tr> <tr> <td style="text-align: center;">2.</td> <td>add_months(d,n) where d-date, n-number of months to be added to the date.</td> <td>Return date after addingthe number of months specified with the function.</td> </tr> <tr> <td style="text-align: center;">3.</td> <td>next_day(d,char)where d-date, char-day of week.</td> <td>Return the date of the first weekday named 'char' that is later than date 'd'.</td> </tr> <tr> <td style="text-align: center;">4.</td> <td>Last_day(d)Where d-date.</td> <td>Return the last day of the month that contain date 'd'. Used to determine how many days left in a month.</td> </tr> <tr> <td style="text-align: center;">5.</td> <td>round(date,[fmt]) where 'fmt'-format model: Month,Day,Year</td> <td>Return date rounded to the unit specified by the format model 'fmt'. If the format model 'fmt' is omitted, date is rounded to the nearest date.</td> </tr> <tr> <td style="text-align: center;">6.</td> <td>trunc(date,[fmt]) where 'fmt'-format model: Month, Day, Year</td> <td>Return date with the time portion of the day truncated to the unit specified by the format model fmt. If the format model fmt is omitted, date is truncated to the nearest day.</td> </tr> </tbody> </table>	Sr.No.	Format	Description	1.	Month_between(d1,d2)where d1 and d2 are dates.	Months_between finds the number of months between d1 and d2. If date d1 is later than d2 the result is positive. If date d1 is earlier than d2 the result is negative.	2.	add_months(d,n) where d-date, n-number of months to be added to the date.	Return date after addingthe number of months specified with the function.	3.	next_day(d,char)where d-date, char-day of week.	Return the date of the first weekday named 'char' that is later than date 'd'.	4.	Last_day(d)Where d-date.	Return the last day of the month that contain date 'd'. Used to determine how many days left in a month.	5.	round(date,[fmt]) where 'fmt'-format model: Month,Day,Year	Return date rounded to the unit specified by the format model 'fmt'. If the format model 'fmt' is omitted, date is rounded to the nearest date.	6.	trunc(date,[fmt]) where 'fmt'-format model: Month, Day, Year	Return date with the time portion of the day truncated to the unit specified by the format model fmt. If the format model fmt is omitted, date is truncated to the nearest day.	<p><i>Any 2 date time function s 1M each</i></p>
Sr.No.	Format	Description																						
1.	Month_between(d1,d2)where d1 and d2 are dates.	Months_between finds the number of months between d1 and d2. If date d1 is later than d2 the result is positive. If date d1 is earlier than d2 the result is negative.																						
2.	add_months(d,n) where d-date, n-number of months to be added to the date.	Return date after addingthe number of months specified with the function.																						
3.	next_day(d,char)where d-date, char-day of week.	Return the date of the first weekday named 'char' that is later than date 'd'.																						
4.	Last_day(d)Where d-date.	Return the last day of the month that contain date 'd'. Used to determine how many days left in a month.																						
5.	round(date,[fmt]) where 'fmt'-format model: Month,Day,Year	Return date rounded to the unit specified by the format model 'fmt'. If the format model 'fmt' is omitted, date is rounded to the nearest date.																						
6.	trunc(date,[fmt]) where 'fmt'-format model: Month, Day, Year	Return date with the time portion of the day truncated to the unit specified by the format model fmt. If the format model fmt is omitted, date is truncated to the nearest day.																						
	<p><b>(d)</b> Ans.</p>	<p><b>With the help of example, explain DROP VIEW command.</b> DROP VIEW statement: Drops the specified view.</p> <p><i>Syntax:</i> <b>DROP VIEW</b> view-Name;</p> <p><i>Example:</i> <b>DROP VIEW emp-view;</b></p> <p><b>Explanation:</b> Any statements referencing the view are invalidated on a DROP VIEW statement. DROP VIEW is disallowed if there are any views or open cursors dependent on the view. The view must be dropped before any objects that it is dependent on can be dropped.</p>	<p><b>4M</b></p> <p><i>Explana tion 2M</i></p> <p><i>Examp le 2M</i></p>																					
	<p><b>(e)</b> Ans.</p>	<p><b>Write a PL/SQL program to print numbers from 50 to 60 using for loop.</b> (Note: Any other Logic also considered)</p> <p>DECLARE x number :=50;</p>	<p><b>4M</b></p> <p><i>Correct program 4M</i></p>																					



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

		<pre>BEGIN   LOOP dbms_output.put_line(x);   x := x +1;   IF x &gt;60 THEN exit; END IF; END LOOP; END;</pre>	
(f) Ans.	<p><b>List and explain types of DBMS users.</b></p> <p><b>List of DBMS user:</b></p> <ol style="list-style-type: none"><li>Naive users</li><li>Application programmers</li><li>Sophisticated users</li><li>Specialized users</li><li>Database administrator</li></ol> <p><b>Explanation:</b></p> <p><b>a) Naive User:</b></p> <ul style="list-style-type: none"><li>Naïve users are unsophisticated users</li><li>They are interact with the system through the application program.</li><li>They give data as input through application program or get output data which is generated by application program.</li><li>Example: Bank cashier.</li></ul> <p><b>b) Application programmers:</b></p> <ul style="list-style-type: none"><li>Application programmers are the users who write the program.</li><li>These programmers use programming tools to develop the program.</li><li>RAD technology is used to write the program.</li></ul> <p><b>c) Sophisticated users:</b></p> <ul style="list-style-type: none"><li>Sophisticated users interact with the system by making the requests in the form of query language.</li><li>These queries are then submitted to the query processor.</li><li>Query processor converts the DML statements into lower</li></ul>	4M  <i>List</i> <i>1M</i>  <i>Explana</i> <i>tion of</i> <i>any 3</i> <i>1M each</i>	





MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

		<p>statement.</p> <p>Whenever a DML statement (INSERT, UPDATE and DELETE) is issued, an implicit cursor is associated with this statement. For INSERT operations, the cursor holds the data that needs to be inserted. For UPDATE and DELETE operations, the cursor identifies the rows that would be affected.</p> <p>In PL/SQL, implicit cursor as has the attributes like %FOUND, %ISOPEN, %NOTFOUND, and %ROWCOUNT.</p> <p><b>Example of implicit cursor:</b> Begin Update emp set salary= salary +500 where empno =&amp;empno; If SQL%FOUND then Dbms_out.put_line(—Emp table modified!); Else Dbms_out.put_line(—Emp table modified!); End if; End;</p> <p><b>Explicit cursor:</b> Explicit cursors are programmer defined cursors for gaining more control over the <b>context area</b>. An explicit cursor should be defined in the declaration section of the PL/SQL Block. It is created on a SELECT Statement which returns more than one row.</p> <p><b>Working with an explicit cursor involves four steps:</b> Declaring the cursor for initializing in the memory <b>Cursor cursor_name IS select_statement;</b> Opening the cursor for allocating memory <b>Open cursorname;</b> Fetching the cursor for retrieving data <b>Fetch cursorname INTO variable1,variable2...</b> Closing the cursor to release allocated memory <b>Close cursorname;</b></p> <p><b>Example of explicit cursor:</b> Declare Cursor c1 is select empno, salary from emp Where deptno=10;</p>	<p><i>Description or example 2M for each type</i></p>
--	--	--	---



**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code: 17332**

		<pre> code emp.empno% Type; sal emp.salary% Type; Begin Open c1; If c1% ISOPEN then Loop Fetch c1 into ecode,sal; If c1% NOTFOUND then Exit; End if; Update emp set salary = salary+500; End Loop; Close c1; Else dbms_out.put_line(—unable to open); End if; End; </pre>	
(c) Ans.	<p><b>Explain strong and weak entity set with example.</b></p> <p><b>Strong Entity Set:</b> An entity set that has sufficient attributes to form a primary key is called as strong entity set.</p> <p><i>Example:</i> Cust_id is a primary key of Customer Entity so customer is strong entity set..</p> <p><b>Weak Entity set:</b> An entity set that does not have sufficient attribute to form a primary key is called as Weak Entity Set.</p> <p><i>Example:</i> transaction is not having sufficient attribute to form a primary key so, transaction a weak entity set</p>	<p><b>4M</b></p> <p><i>Each entity description on 1M</i></p> <p><i>Example 1M each (diagram optional)</i></p>	



**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code:**

<b>17332</b>
--------------

<b>(d)</b> Ans.	<p><b>Differentiate between view and index.</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Parameter</th> <th style="width: 35%;">View</th> <th style="width: 45%;">Index</th> </tr> </thead> <tbody> <tr> <td><b>Definition</b></td> <td>A view is a virtual table, through which a selective portion of data from one or more tables can be seen</td> <td>An index is a schema object that can speed up the retrieval of rows by using pointer.</td> </tr> <tr> <td><b>Function</b></td> <td>View restrict access to data such that user can see and modify exactly what they need and no more.</td> <td>Index provides direct and fast access to rows in a table</td> </tr> <tr> <td><b>Syntax</b></td> <td>CREATE VIEW view_name AS SELECT column1, column2..... FROM table_name WHERE [condition];</td> <td>Create index index_name on Table_name (column_name);</td> </tr> <tr> <td><b>Example</b></td> <td>CREATE VIEW CUSTOMERS_VIEW AS SELECT name, age FROM CUSTOMERS;</td> <td>create index s_index on Student(roll_no);</td> </tr> </tbody> </table>	Parameter	View	Index	<b>Definition</b>	A view is a virtual table, through which a selective portion of data from one or more tables can be seen	An index is a schema object that can speed up the retrieval of rows by using pointer.	<b>Function</b>	View restrict access to data such that user can see and modify exactly what they need and no more.	Index provides direct and fast access to rows in a table	<b>Syntax</b>	CREATE VIEW view_name AS SELECT column1, column2..... FROM table_name WHERE [condition];	Create index index_name on Table_name (column_name);	<b>Example</b>	CREATE VIEW CUSTOMERS_VIEW AS SELECT name, age FROM CUSTOMERS;	create index s_index on Student(roll_no);	<p><b>4M</b></p> <p><i>Any 2 relevant points : 2M Each</i></p>
Parameter	View	Index															
<b>Definition</b>	A view is a virtual table, through which a selective portion of data from one or more tables can be seen	An index is a schema object that can speed up the retrieval of rows by using pointer.															
<b>Function</b>	View restrict access to data such that user can see and modify exactly what they need and no more.	Index provides direct and fast access to rows in a table															
<b>Syntax</b>	CREATE VIEW view_name AS SELECT column1, column2..... FROM table_name WHERE [condition];	Create index index_name on Table_name (column_name);															
<b>Example</b>	CREATE VIEW CUSTOMERS_VIEW AS SELECT name, age FROM CUSTOMERS;	create index s_index on Student(roll_no);															
<b>(e)</b> Ans.	<p><b>Explain the concept of trigger.</b></p> <p>A trigger is a PL/SQL block structure which is fired when DML statements like Insert, Delete, Update is executed on a database table. A trigger is triggered automatically when an associated DML statement is executed.</p> <p><b>Syntax for Creating a Trigger:</b> CREATE OR REPLACE TRIGGER trigger_name [BEFORE/AFTER] [INSERT/UPDATE/DELETE] ON Table_name [FOR EACH STATEMENT/FOR EACH ROW] [WHEN CONDITION] PL/SQL block</p> <p><b>Example:</b> CREATE OR REPLACE TRIGGER trg1</p>	<p><b>4M</b></p> <p><i>Descript ion 4M (Examp le Optiona l)</i></p>															



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

		BEFORE INSERT ON EMP FOR EACH ROW BEGIN IF :new.sal<=0 THEN Raise_application_error('Salary should be greater than 0'); END IF; END;	
(f) Ans.	<b>Describe function of database administrator.</b> <b>Function of database administrator</b> <b>1.Schema Definition</b> The Database Administrator creates the database schema by executing DDL statements. Schema includes the logical structure of database table (Relation) like data types of attributes, length of attributes, integrity constraints etc. <b>2. Storage structure and access method definition</b> The DBA creates appropriate storage structures and access methods by writing a set of definitions which is translated by data storage and DDL compiler. <b>3. Schema and physical organization modification</b> DBA writes set of definitions to modify the database schema or description of physical storage organization. <b>4. Granting authorization for data access</b> The DBA provides different access rights to the users according to their level. Ordinary users might have highly restricted access to data, while you go up in the hierarchy to the administrator, you will get more access rights. Integrity constraints specifications: Integrity constraints are written by DBA and they are stored in a special file which is accessed by database manager while updating data. <b>5. Routine Maintenance</b> some of the routine maintenance activities of a DBA is given below. (i) Taking backup of database periodically (ii) Ensuring enough disk space is available all the time. (iii) Monitoring jobs running on the database. (iv) Ensure that performance is not degraded by some expensive task submitted by some users.	<b>4M</b>  <i>Any 4 functions :1M each</i>	



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

		<p><b>6. Integrity- constraint specification:</b> Integrity constraints are written by DBA and they are stored in a special file, which is accessed by database manager, while updating the data.</p>	
4.	(a)  Ans.	<p><b>Attempt any FOUR of the following:</b> <b>Draw E-R diagram for customer and loan. Assume suitable attribute.</b></p> <p><b>OR</b></p> <p><i>Correct use of symbols 2M</i></p> <p><i>Correct entities and relation ships 2M</i></p>	16 4M



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

	<p>(b)</p> <p>Ans.</p>	<p><b>Define following term with example.</b></p> <p><b>(i) Procedure</b> <b>(ii) Function</b></p> <p><b>(i) Procedure:</b> Definition: A procedure is named PL/SQL block which perform one or more specifies task.</p> <p><i>Example</i></p> <p>The following example creates a simple procedure that displays the string 'Hello World!' on the screen when executed.</p> <pre>CREATE OR REPLACE PROCEDURE greetings AS BEGIN dbms_output.put_line('Hello World!'); END;</pre> <p><b>(ii) Function:</b></p> <p>Definition: Function is a logically grouped set of SQL and PI/SQL statements that perform a specific task.</p> <p><i>Example:</i></p> <p>This function returns the total number of CUSTOMERS in the customers table.</p> <pre>CREATE OR REPLACE FUNCTION totalCustomers RETURN number IS total number(2):=0; BEGIN SELECT count(*)into total FROM customers;  RETURN total; END;</pre> <p>/</p>	<p>4M</p> <p><i>Each term :Definition on 1M, Example: 1M</i></p>
--	------------------------	---	--







**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code: 17332**

		<p>embedded in application program to normal procedural calls in host language. The precompiler must interact with the DML compiler to generate the appropriate code.</p> <p>4. <b>Query Evaluation Engine:</b> It executes low level instructions generated by DML compiler and DDL</p>	
5.	<p>(a) Ans.</p>	<p><b>Attempt any FOUR of the following:</b>  <b>Explain three tier architecture.</b>          In three tier architecture, the communication taken place from client to application server and then application server to database system to access the data. The application server or web server is sometimes called middle layer or intermediate layer. The middle layer which processes applications and database server processes the queries. This type of communication system is used in the large applications or the world web applications. On WWW all clients requests for data and server serves it. There are multiple servers used like fax server, proxy server, mail server etc.</p> <div style="text-align: center;"> <pre>           graph TD             subgraph Client               GUI[GUI, Web Interface]             end             subgraph AS[Application Server or Web Server]               AP[Application Programs, Web Pages]             end             subgraph DS[Database Server]               DBMS[Database Management System]             end             GUI &lt;--&gt; AP             AP &lt;--&gt; DBMS           </pre> </div>	<p><b>16 4M</b></p> <p><i>Explanation 4M</i></p> <p><i>(Diagram optional)</i></p>
	<p>(b) Ans.</p>	<p><b>Explain snapshot with example.</b>  <b>Snapshot:</b>          It is also known as materialized view.          It is a copy of either an entire single table or set of its rows or collection of tables, views or rows using join, grouping and selection criteria.          Useful in distributed environment  <b>It has two types:</b>          Simple snapshot and complex snapshot. Simple snapshot related to single table and complex snapshot related to joined</p>	<p><b>4M</b></p> <p><i>Snapshot explanation 2M</i></p>



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

		<p>tables.</p> <p><b>Example :</b></p> <p><b>Operations on snapshot:</b></p> <p><b>i) Creating Snapshot:</b> Create snapshot command is used to create the snapshot. <i>Syntax:-</i> CREATE SNAPSHOT [schema.] &lt;snapshot name&gt;AS subquery; <i>Example:-</i> Create snapshot emp_snapas select * from emp where deptno=6;</p> <p><b>ii) Altering snapshot:</b> Snapshot can be altered by using ALTER SNAPSHOT command. The only parts of a snapshot that can be altered are its storage parameters, refresh type and refresh start, and next interval. The select for the snapshot, base tables, and other data related items cannot be changed without dropping and recreating the snapshot. <i>Syntax:-</i> ALTER SNAPSHOT &lt;snapshotname&gt; [[PCTFREE &lt;integer&gt;] [PCTUSED &lt;integer&gt;] [REFRESH [FAST/COMPLETE/FORCE]]]; <i>Example:-</i> To change the automatic refresh mode for the emp_data snapshot to fast: ALTER SNAPSHOT emp_data REFRESH FAST;</p> <p><b>iii) Dropping a snapshot</b> To remove the snapshot DROP SNAPSHOT Command .When snapshot is dropped which a snapshot log had associated with it, only the rows required for maintaining that snapshot are dropped. <i>Syntax:-</i> Drop snapshot &lt;snapshot name&gt;; <i>Example:-</i> Drop snapshot emp_snap;</p>	<p><i>Any one operation example :2M</i></p>
--	--	--	---



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

	<p>(c) Ans.</p>	<p><b>Explain GOTO statement with example.</b> The GOTO statement transfers control to a labelled block or statement. It is an unconditional branching statement. It does not use any condition for transferring the control to other part of code. It transfers the control to the part of code which contains same label mentioned in goto statement. <i>Syntax :</i> GOTO label; .. .. &lt;&lt; label &gt;&gt; statement;  <i>Example :</i> DECLARE   p VARCHAR2(30);   n PLS_INTEGER := 37; BEGIN   FOR j in 2..ROUND(SQRT(n)) LOOP     IF n MOD j = 0 THEN       p := 'is not a prime number';       GOTO print_now;     END IF;   END LOOP;   p := 'is a prime number'; &lt;&lt;print_now&gt;&gt;   DBMS_OUTPUT.PUT_LINE(TO_CHAR(n)    p); END;</p>	<p>4M  <i>Explanation</i> 2M  <i>Example</i> 2M (any other example considered)</p>
	<p>(d) Ans.</p>	<p><b>Explain PL/SQL block structure.</b> <b>Block structure of PL/SQL:</b> Declare      <i>Declaration of memory variables</i>  BEGIN (Mandatory)      <i>SQL executable statements</i>  Exception</p>	<p>4M  <i>Correct block structure</i> 3M</p>



MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION  
(Autonomous)  
(ISO/IEC - 27001 - 2005 Certified)

MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

		<p><i>Handling errors</i></p> <p>END; (Mandatory)</p> <p>A block begins with a declarative section where variables are declared. This is followed by a section containing the procedural statements surrounded by BEGIN and END keywords. Each block must have Begin and End statements and may optionally include an exception section to handle errors. End section marks the end of PL-SQL block</p>	<p><i>Explanation 1M</i></p>
(e) Ans.	<p><b>Explain candidate key and primary key.</b></p> <p><b>Candidate key:</b> In a relation, there may be a primary key or may not, but there may be a key or combination of keys which uniquely identify the record. Such a key is called as Candidate key.</p> <p><b>Primary key:</b> A key which is selected by the designer to uniquely identify the entity is called as Primary key. A primary key cannot contain duplicate values and it can never contain null values inside it.</p>	<p>4M</p> <p>2M each</p>	
(f) Ans.	<p><b>Explain concurrent execution of multiple transaction in detail.</b></p> <p>A transaction is a collection of operations that perform modifications in database application.</p> <p>A set of transactions can be executed by some sequence which is called as a schedule.</p> <p>There are two types of schedules as serial executions and concurrent execution.</p> <p>When two transactions are executed one after the other, it is called as serial execution and when they are interleaved, it is called as concurrent execution of transaction.</p> <p>Concurrent execution should always preserve consistency. This means that even if the transactions are interleaved, there should not be any damage to data.</p> <p><i>Example :</i> Transaction T1 :Rs. 50 of A's account are transferred to B's Account. Transaction T2 : 10% of A's balance is transferred to B's Account Consider initial amount as A=100, B=150 so initially A+B=150 Concurrent schedule will appear as :</p>	<p>4M</p> <p><i>Explanation 4M</i></p>	



**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code: 17332**

		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">T1</td> <td style="width: 50%;">T2</td> </tr> <tr> <td>Read(A); .....A=100</td> <td></td> </tr> <tr> <td>A:=A-50;</td> <td></td> </tr> <tr> <td>Write(A); .....A=50</td> <td></td> </tr> <tr> <td></td> <td>Read(A); .....A=50</td> </tr> <tr> <td></td> <td>Temp:=A*0.1;</td> </tr> <tr> <td></td> <td>A:=A-temp;</td> </tr> <tr> <td></td> <td>Write(A); .....A=45</td> </tr> <tr> <td>Read(B); .....B=150</td> <td></td> </tr> <tr> <td>B:=B+50;</td> <td></td> </tr> <tr> <td>Write(B); .....B=200</td> <td></td> </tr> <tr> <td></td> <td>Read(B); .....B=200</td> </tr> <tr> <td></td> <td>B:=B+temp;</td> </tr> <tr> <td></td> <td>Write(B); .....B=205</td> </tr> <tr> <td></td> <td></td> </tr> </table> <p>Initially A=100, B=150 therefore A+B=250.</p> <p>At the end of both the transactions in above concurrent schedule. The consistency is preserved. i.e., A=45, B=205 and therefore A+B=250</p>	T1	T2	Read(A); .....A=100		A:=A-50;		Write(A); .....A=50			Read(A); .....A=50		Temp:=A*0.1;		A:=A-temp;		Write(A); .....A=45	Read(B); .....B=150		B:=B+50;		Write(B); .....B=200			Read(B); .....B=200		B:=B+temp;		Write(B); .....B=205			
T1	T2																																
Read(A); .....A=100																																	
A:=A-50;																																	
Write(A); .....A=50																																	
	Read(A); .....A=50																																
	Temp:=A*0.1;																																
	A:=A-temp;																																
	Write(A); .....A=45																																
Read(B); .....B=150																																	
B:=B+50;																																	
Write(B); .....B=200																																	
	Read(B); .....B=200																																
	B:=B+temp;																																
	Write(B); .....B=205																																
<b>6.</b>	<p><b>(a)</b> Ans.</p>	<p><b>Attempt any FOUR of the following:</b>  <b>Explain client-Server architecture.</b>          Client server is a system in which all the data &amp; information is stored on the server &amp; all application programs are stored on the client. In this client server relationship, several clients can share the same server.</p> <div style="text-align: center;"> <pre>           graph TD             C1[CLIENT 1] --- N[NETWORK]             C2[CLIENT 2] --- N             Cn[CLIENT n] --- N             N --- S[SERVER]           </pre> </div> <p>Computer networking allows some task to be executed on a server system and some tasks on client system. This leads to development of client server architecture. The clients are the machines which requests for the service to the server. Server is the machine which serves to the</p>	<p style="text-align: center;"><b>16</b> <b>4M</b></p> <p style="text-align: center;"><i>Explanation 4M</i></p> <p style="text-align: center;"><i>Diagram Optional</i></p>																														



*MODEL ANSWER*

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code: 17332**

		<p>clients. There are different types of client/server architecture such as</p> <ul style="list-style-type: none"> <li>• Two tier architecture,</li> <li>• Three tier architecture.</li> </ul> <p>In two tier architecture, client systems directly approach database servers whereas in three tier architecture, there exists a middle layer which acts as application server to receive and send requests from client machine to database server and vice versa</p>																
(b) Ans.		<p><b>Explain Third normal form.</b></p> <ul style="list-style-type: none"> <li>• <b>3NF (Third normal form)</b> <ol style="list-style-type: none"> <li>1) After removing all transitive dependencies and making separate relations, relations get into 3NF.</li> <li>2) Transitive dependency is can be stated as, let R be relation and A, B and C be the set of attributes then, they are transitive dependent if C depends on B, B depends on A and therefore C depends on A.</li> </ol> </li> </ul> <p><i>Example</i></p> <p><b>Student_Detail Table :</b></p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <th>Student_id</th> <th>Student_name</th> <th>DOB</th> <th>Street</th> <th>city</th> <th>State</th> <th>Zip</th> </tr> </table> <p>In this table Student_id is Primary key, but street, city and state depends upon Zip. The dependency between zip and other fields is called <b>transitive dependency</b>. Hence to apply <b>3NF</b>, we need to move the street, city and state to new table, with <b>Zip</b> as primary key.</p> <p><b>New Student_Detail Table :</b></p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <th>Student_id</th> <th>Student_name</th> <th>DOB</th> <th>Zip</th> </tr> </table> <p><b>Address Table :</b></p> <table border="1" style="width: 100%; text-align: center; border-collapse: collapse;"> <tr> <th>Zip</th> <th>Street</th> <th>city</th> <th>State</th> </tr> </table>	Student_id	Student_name	DOB	Street	city	State	Zip	Student_id	Student_name	DOB	Zip	Zip	Street	city	State	<p><b>4M</b></p> <p><i>Explanation 3M</i></p> <p><i>Example 1M</i></p>
Student_id	Student_name	DOB	Street	city	State	Zip												
Student_id	Student_name	DOB	Zip															
Zip	Street	city	State															

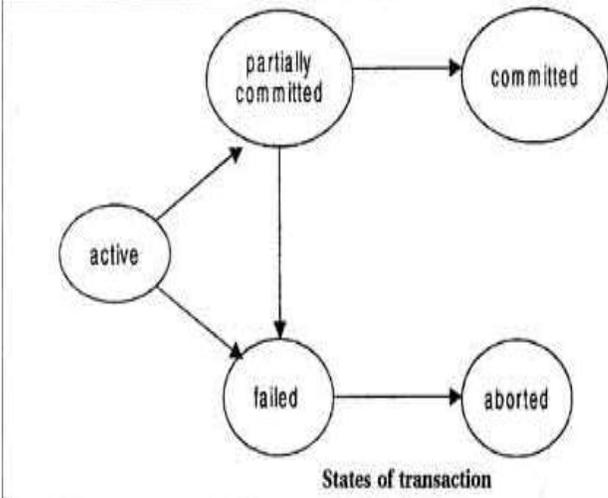


MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

<p>(c) Ans.</p>	<p>Describe state of transaction with neat diagram.</p>  <p>States of transaction</p> <p>A transaction must be in one of the following states:</p> <ol style="list-style-type: none"><li>1. <b>Active:</b> the initial state, the transaction stays in this state while it is executing.</li><li>2. <b>Partially committed:</b> after the final statement has been executed.</li><li>3. <b>Failed:</b> when the normal execution can no longer proceed.</li><li>4. <b>Aborted:</b> after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction.</li><li>5. <b>Committed:</b> after successful completion.</li></ol>	<p>4M</p> <p><i>Diagram 2M</i></p> <p><i>Explanation 2M</i></p>
---------------------	---	---



**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code:**

<b>17332</b>
--------------

<p>(d)  Ans.</p>	<p><b>Draw E-R diagram for airline reservation system. Here a passenger can book ticket from personal for a flight on same date.</b></p> <div style="text-align: center; margin: 20px 0;"> </div>	<p style="text-align: center;"><b>4M</b></p> <p style="text-align: center;"><i>Correct use of symbols 2M</i></p> <p style="text-align: center;"><i>Correct entities and relationships 2M</i></p>								
<p>(e)  Ans.</p>	<p><b>Describe serializability.</b></p> <p>In concurrent execution of transaction, if the consistency level of the concurrent schedule is same as the consistency level after serial schedule of the same schedule, then that concurrent schedule is called as serializable concurrent schedule and this property of schedule is called as serializability.</p> <p>Serializability ensures consistency of database.</p> <p><i>Example :</i></p> <p>Transaction T1: Rs. 50 of A's account are transferred to B's Account.        Transaction T2 : 10% of A's balance is transferred to B's Account        Consider initial amount as A=100, B=150 so initially A+B=250        Serial Schedule appears as</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 50%;">T1</th> <th style="width: 50%;">T2</th> </tr> </thead> <tbody> <tr> <td>Read(A); .....A=100</td> <td></td> </tr> <tr> <td>A:=A-50;</td> <td></td> </tr> <tr> <td>Write(A);</td> <td>.....A=50</td> </tr> </tbody> </table>	T1	T2	Read(A); .....A=100		A:=A-50;		Write(A);	.....A=50	<p style="text-align: center;"><b>4M</b></p> <p style="text-align: center;"><i>Description with suitable example 4M</i></p>
T1	T2									
Read(A); .....A=100										
A:=A-50;										
Write(A);	.....A=50									



**MAHARASHTRA STATE BOARD OF TECHNICAL EDUCATION**  
 (Autonomous)  
 (ISO/IEC - 27001 - 2005 Certified)

**MODEL ANSWER**

**SUMMER - 2017 EXAMINATION**

**Subject: Relational Database Management System**

**Subject Code:**

<b>17332</b>
--------------

Read(B); .....B=150	
B:=B+50;	
Write(B); .....B=200	
	Read(A); .....A=50
	Temp:=A*0.1;
	A:=A-temp;
	Write(A); .....A=45
	Read(B); .....B=200
	B:=B+temp;
	Write(B); .....B=205

At the end of serial schedule,  $A+B=250$   
 Concurrent schedule will appear as:

T1	T2
Read(A); .....A=100	
A:=A-50;	
Write(A); .....A=50	
	Read(A); .....A=50
	Temp:=A*0.1;
	A:=A-temp;
	Write(A); .....A=45
Read(B); .....B=150	
B:=B+50;	
Write(B); .....B=200	
	Read(B); .....B=200
	B:=B+temp;
	Write(B); .....B=205

Here also  $A+B=250$  at the end of schedule, so this concurrent schedule preserves consistency and hence it is a serializable schedule and shows serializability property



MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System

Subject Code: 17332

	<p>(f) Ans.</p>	<p><b>Explain locking strategy in detail.</b></p> <p><b>Implicit Locking:</b> Implicit locks are generally placed by the DBMS automatically.</p> <p><b>Explicit Locking:</b> The technique of lock taken on a table or its resources by a user is called Explicit Locking. Users can lock tables they own or any tables on which they have been granted table privileges (such as select, insert, update, delete). Explicit locking done by two ways as</p> <p><b>1) The Select.....For Update statement</b> It is used for acquiring exclusive row level locks in anticipation of performing updates on records.</p> <p><b>2) Using lock table statement:</b> To manually override Oracle's default locking strategy by creating a data lock in a specific mode.</p> <p><b>Syntax:</b></p> <p><b>LOCK TABLE &lt;TableName&gt; [, &lt;TableName&gt;]...</b> <b>IN { ROW SHARE  ROW EXCLUSIVE SHARE</b> <b>UPDATE SHARE SHARE ROW EXCLUSIVE EXCLUSIVE}</b> <b>[NOWAIT]</b></p>	<p>4M</p> <p><i>Explanation 4M</i></p>
--	---------------------	---	--