



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

SUMMER – 2019 EXAMINATION  
MODEL ANSWER

Subject: Database Management

Subject Code: 22416

**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) Ans.	<b>Attempt any <u>FIVE</u> of the following: Enlist any four relational algebra operators.</b> The fundamental operators of relational algebra are as follows – i. Select ( $\sigma$ ) ii. Project ( $\Pi$ ) iii. Union Operation ( $\cup$ ) iv. Set Difference ( $-$ ) v. Cartesian Product ( $\times$ ) vi. Rename Operation ( $\rho$ )	<b>10 2M</b>  <i>Any four operator s <sup>1/2</sup>M each</i>
	b) Ans.	<b>State the use of 'Between' clause (any two user)</b>  The SQL BETWEEN clause allows user to easily test if an expression is within a range of values (inclusive). The values can be text, date, or numbers. It can be used in a SELECT, INSERT, UPDATE, or DELETE statement. The SQL BETWEEN Condition will return the records where expression is within the range of value1 and value2 inclusive of the the values.	<b>2M</b>  <i>Any two uses 1M each</i>



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

<b>c)</b>	<b>Ans.</b>	<p><b>Compare ‘GROUP BY’ and ‘HAVING’ clause. (any two difference).</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%; text-align: center;">Sr. No.</th> <th style="width: 40%; text-align: center;">GROUP BY</th> <th style="width: 50%; text-align: center;">HAVING</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>The GROUP BY Clause is utilized in SQL with the SELECT statement to organize similar data into groups. It combines the multiple records in single or more columns using aggregate functions.</td> <td>HAVING Clause is utilized in SQL as a conditional Clause with GROUP BY Clause.</td> </tr> <tr> <td style="text-align: center;">2</td> <td>GROUP BY returns only one result per group of data.</td> <td>This conditional clause returns rows where aggregate function results match with given conditions only.</td> </tr> </tbody> </table>	Sr. No.	GROUP BY	HAVING	1	The GROUP BY Clause is utilized in SQL with the SELECT statement to organize similar data into groups. It combines the multiple records in single or more columns using aggregate functions.	HAVING Clause is utilized in SQL as a conditional Clause with GROUP BY Clause.	2	GROUP BY returns only one result per group of data.	This conditional clause returns rows where aggregate function results match with given conditions only.	<p><b>2M</b></p> <p style="text-align: center;"><i>Any two differences 1M each</i></p>
Sr. No.	GROUP BY	HAVING										
1	The GROUP BY Clause is utilized in SQL with the SELECT statement to organize similar data into groups. It combines the multiple records in single or more columns using aggregate functions.	HAVING Clause is utilized in SQL as a conditional Clause with GROUP BY Clause.										
2	GROUP BY returns only one result per group of data.	This conditional clause returns rows where aggregate function results match with given conditions only.										
<b>d)</b>	<b>Ans.</b>	<p><b>State any two advantages of PL/SQL.</b></p> <p><b>Advantages of PL/SQL are:</b></p> <ul style="list-style-type: none"> <li>SQL is the standard database language and PL/SQL is strongly integrated with SQL. PL/SQL supports both static and dynamic SQL. Static SQL supports DML operations and transaction control from PL/SQL block. In Dynamic SQL, SQL allows embedding DDL statements in PL/SQL blocks.</li> <li>PL/SQL allows sending an entire block of statements to the database at one time. This reduces network traffic and provides high performance for the applications.</li> <li>PL/SQL gives high productivity to programmers as it can query, transform, and update data in a database.</li> <li>PL/SQL saves time on design and debugging by strong features, such as exception handling, encapsulation, data hiding, and object-oriented data types.</li> <li>Applications written in PL/SQL are fully portable.</li> <li>PL/SQL provides high security level.</li> </ul>	<p><b>2M</b></p> <p style="text-align: center;"><i>Any two advantages 1M each</i></p>									
<b>e)</b>	<b>Ans.</b>	<p><b>List any four statements of PL/SQL.</b></p> <p>PL/SQL has different categories of control statements. They are the following:</p> <ul style="list-style-type: none"> <li><b>Conditional selection statements:</b> Includes IF statement and Case statement</li> </ul>	<p><b>2M</b></p> <p style="text-align: center;"><i>Any four statements <sup>1/2</sup>M each</i></p>									



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<ul style="list-style-type: none"> <li>• <b>IF statement</b> has these forms:               <ul style="list-style-type: none"> <li>• IF THEN</li> <li>• IF THEN ELSE</li> <li>• IF THEN ELSIF</li> </ul> </li>   <li>• <b>Loop statements:</b>                The loop statements are the basic LOOP, FOR LOOP, and WHILE LOOP</li>   <li>• <b>Sequential control statements,</b> : EXIT, CONTINUE and GOTO Statements are Sequential Control Statements.</li> </ul>	
	<p><b>f) Ans.</b></p>	<p><b>Define the term ‘cursor’.</b>          Oracle creates a memory area, known as the context area, for processing an SQL statement, which contains all the information needed for processing the statement; for example, the number of rows processed, etc.          A cursor is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the active set.</p>	<p><b>2M</b></p> <p><i>Definition 2M</i></p>
	<p><b>g) Ans.</b></p>	<p><b>Enlist the types of database users.</b>  <b>Different Types of Database Users in DBMS:</b>          1. Application Programmers          2. End Users         <ul style="list-style-type: none"> <li>• Casual User</li> <li>• Naive</li> <li>• Sophisticated users</li> <li>• Standalone users</li> </ul>         3. DBA (Database Administrator)          4. System Analyst</p>	<p><b>2M</b></p> <p><i>Each type <sup>1/2</sup>M</i></p>
<p><b>2.</b></p>	<p><b>a) Ans.</b></p>	<p><b>Attempt any <u>THREE</u> of the following:</b>  <b>Describe simple and composite attribute with suitable example.</b>  <b>Simple attribute</b> – Simple attributes are atomic values, which cannot be divided further.          For example, a student's phone number is an atomic value of 10 digits.  <b>Composite attribute</b> – Composite attributes are made of more than one simple attribute.</p>	<p><b>12</b> <b>4M</b></p> <p><i>Each attribute with example 2M</i></p>



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		For example, a student's complete name may have first_name and last_name.													
	<b>b) Ans.</b>	<p><b>Define the index with its characteristics.</b></p> <p><b>Definition</b> An index is a schema object. It is used by the oracle server to speed up the retrieval of rows by using a pointer.</p> <p><b>Characteristics of index:</b></p> <ol style="list-style-type: none"> <li>1. It can reduce disk i/o by using a rapid path access to locate data quickly.</li> <li>2. Indexes are logically and physically independent of the table they index.</li> <li>3. These are maintained automatically by the oracle server.</li> </ol>	<p><b>4M</b></p> <p><i>Definitio n 2M</i></p> <p><i>Any two characte ristics 1M each</i></p>												
	<b>c) Ans.</b>	<p><b>Differentiate between PL/SQL function and procedure. (any four differences).</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%; text-align: center;">Function</th> <th style="width: 50%; text-align: center;">Procedure</th> </tr> </thead> <tbody> <tr> <td>1.The function must return a value</td> <td>1. In Stored Procedure value return is optional. Even a procedure can return zero or n values.</td> </tr> <tr> <td>2. Functions can have only input parameters for it</td> <td>2. Procedures can have input or output parameters.</td> </tr> <tr> <td>3. Functions can be called from Procedure</td> <td>3. Procedures cannot be called from a Function.</td> </tr> <tr> <td>4. Function allows only SELECT statement in it.</td> <td>4. The procedure allows SELECT as well as DML (INSERT/UPDATE/DELETE) statement in it.</td> </tr> <tr> <td>5. Try-catch block cannot be used in a Function.</td> <td>5. An exception can be handled by try-catch block in a Procedure</td> </tr> </tbody> </table>	Function	Procedure	1.The function must return a value	1. In Stored Procedure value return is optional. Even a procedure can return zero or n values.	2. Functions can have only input parameters for it	2. Procedures can have input or output parameters.	3. Functions can be called from Procedure	3. Procedures cannot be called from a Function.	4. Function allows only SELECT statement in it.	4. The procedure allows SELECT as well as DML (INSERT/UPDATE/DELETE) statement in it.	5. Try-catch block cannot be used in a Function.	5. An exception can be handled by try-catch block in a Procedure	<p><b>4M</b></p> <p><i>Any four differen ces 1M each</i></p>
Function	Procedure														
1.The function must return a value	1. In Stored Procedure value return is optional. Even a procedure can return zero or n values.														
2. Functions can have only input parameters for it	2. Procedures can have input or output parameters.														
3. Functions can be called from Procedure	3. Procedures cannot be called from a Function.														
4. Function allows only SELECT statement in it.	4. The procedure allows SELECT as well as DML (INSERT/UPDATE/DELETE) statement in it.														
5. Try-catch block cannot be used in a Function.	5. An exception can be handled by try-catch block in a Procedure														
	<b>d) Ans.</b>	<p><b>Describe database security with its requirements.</b></p> <p>Database security refers to the collective measures used to protect and</p>	<p><b>4M</b></p>												



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

SUMMER – 2019 EXAMINATION  
MODEL ANSWER

Subject: Database Management

Subject Code: 22416

		<p>secure a database or database management software from illegitimate use and malicious threats and attacks.</p> <p>It is a broad term that includes a multitude of processes, tools and methodologies that ensure security within a database environment. Database security covers and enforces security on all aspects and components of databases. This includes:</p> <ul style="list-style-type: none"><li>• Data stored in database</li><li>• Database server</li><li>• Database management system (DBMS)</li><li>• Other database workflow applications</li></ul> <p><b>Database Security Requirements:</b></p> <p>The basic database security requirements which technology can ensure are the following</p> <ol style="list-style-type: none"><li>1. Confidentiality: A secure system ensures the confidentiality of data. This means that it allows individuals to see only the data they are supposed to see. Confidentiality has several aspects like privacy of communications, secure storage of sensitive data, authenticated users and authorization of users.</li><li>2. Integrity: A secure system ensures that the data it contains is valid. Data is integrated means that data is protected from deletion and corruption, both while it resides within the data-case, and while it is being transmitted over the network.</li><li>3. Availability: A secure system makes data available to authorized users, without delay. Denial of service attacks are attempts to block authorized users' ability to access and use the system when needed.</li></ol>	<p><i>Description 2M</i></p> <p><i>Requirements 2M</i></p>
3.	a) Ans.	<p><b>Attempt any <u>THREE</u> of the following:</b></p> <p><b>Give any four string functions with example.</b></p> <p><b>1. initcap(str)</b> Converts first letter of string to capital letter. Example: Select initcap('rdbms') from dual;</p> <p><b>2. lower(str)</b> Converts a string to all lowercase characters. Example: Select lower('RDBMS') from dual;</p>	<p>12 4M</p> <p><i>Any 4 function 1M each</i></p>



SUMMER – 2019 EXAMINATION  
MODEL ANSWER

Subject: Database Management

Subject Code: 22416

	<p><b>3. upper(str)</b> Converts a string to all uppercase characters. Example: Select upper('rdbms') from dual;</p> <p><b>4. length(str)</b> Find out the length of given string. Example: Select length('RDBMS') from dual;</p> <p><b>5. ltrim(str)</b> Removes leading spaces from a string Example: Select ltrim(' abcd ') from dual;</p> <p><b>6. rtrim(str)</b> Removes trailing spaces from a string Example: Select rtrim(' abcd ') from dual;</p> <p><b>7. Lpad(char1,length,char2)</b> It returns char1, left-padded to given length with the sequence of characters in char2. Example: Select lpad('SKY', 8, '*') from dual;</p> <p><b>8. Rpad(char1,length,char2)</b> It returns char1, right-padded to given length with the sequence of characters in char2. Example: Select rpad ('SKY', 8, '*') from dual;</p> <p><b>9. Translate(char,from string, to string)</b> It returns expr with all occurrences of each character in from_string replaced by its corresponding character in to_string Example: Select translate (Hickory,'H','D') from dual;</p> <p><b>10. Replace(char,searchstring,[repstring])</b></p>	
--	---	--



**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<p>It returns character string with each occurrences of searchstring replaced with [repstring]          Example:          Select replace('Tick and Tock','T','Cl') from dual;</p> <p><b>11. Substr(char,m,n)</b>          It returns substring of character string that start at m character and is of length n          Example:          Select substr(Triangle'4,5) from dual;</p> <p><b>12. Concat(str, str2)</b>          It merges two or more strings or a string and a data value together          Example:          select concat('summer ','18') from dual;</p> <p><b>13. Chr(n)</b>          Returns a character binary equivalent of n.          select chr(65) from dual;</p> <p><b>14. Ascii(char)</b>          Returns a decimal representation of a character.          select ascii ('A') from dual;</p>	
	<p><b>b)</b> <b>Ans.</b></p>	<p><b>Create synonyms for 'class' tables. Write steps to create synonyms.</b>  <b>Create synonyms for class tables.</b>          The following code shows how to create a synonym for the class table          Example:          Create Synonym Class.syn for class;</p> <p><b>Write steps to create synonyms</b>          Syntax:          CREATE [OR REPLACE] [PUBLIC] SYNONYM [schema .]          synonym_name          FOR [schema.] object_name;</p>	<p style="text-align: center;"><b>4M</b></p> <p style="text-align: center;"><i>2M for creating synonym</i></p> <p style="text-align: center;"><i>2M to show the syntax of how to create synonym</i></p>
	<p><b>c)</b> <b>Ans.</b></p>	<p><b>Write a PL/SQL program to display 10 reverse numbers. Use 'for' loop.</b></p>	<p style="text-align: center;"><b>4M</b></p>





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

SUMMER – 2019 EXAMINATION  
MODEL ANSWER

Subject: Database Management

Subject Code: 22416

	<p><b>i) Find customer name having saving account as well as loan account:</b> Select d.Cust_name From depositor d, borrower b Where d. Cust_name=b.cust_name;</p> <p><b>(ii) Find customer names having loan account but not the savings account:</b> Select b.cust_name from borrower minus Select d.cust_name from depositor Where d. Cust_name=b.cust_name; (any other relevant logic can be considered for the query)</p>	<p><i>Each query 2M</i></p>
<p><b>b)</b> <b>Ans.</b></p>	<p><b>Create sequence for department table and also altered the created sequence.</b></p> <p>Create sequence deptid Start with 1 Increment by 1 Maxvalue 100;</p> <p>Alter the created sequence Alter sequence deptid maxvalue 1500;</p>	<p><b>4M</b></p> <p><i>2M any correct syntax of create sequence</i> <i>2M for altering it</i></p>
<p><b>c)</b> <b>Ans.</b></p>	<p><b>List the types of trigger. Write the steps to create trigger with example.</b></p> <p><b>Types of Triggers:</b></p> <ol style="list-style-type: none"><li>1. Row-level trigger</li><li>2. Statement-level trigger</li><li>3. Before-trigger</li><li>4. After-trigger</li></ol> <p>Steps to create trigger: 1) Trigger can be created with the following syntax in database environment :</p> <pre>CREATE [OR REPLACE ] TRIGGER trigger_name {BEFORE   AFTER   INSTEAD OF } {INSERT [OR]   UPDATE [OR]   DELETE} [OF col_name] ON table_name [REFERENCING OLD</pre>	<p><b>4M</b></p> <p><i>2M for listing</i></p> <p><i>2M for steps with example</i></p>



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<p>AS o NEW AS n] [FOR EACH ROW] WHEN (condition)</p> <p>2) User does not have to fire the trigger, but it gets automatically fired according to definition of the trigger.</p> <p>Example :</p> <pre>CREATE OR REPLACE TRIGGER display_salary_changes BEFORE DELETE OR INSERT OR UPDATE ON customers FOR EACH ROW WHEN (NEW.ID &gt; 0) DECLARE     sal_diff number; BEGIN     sal_diff := :NEW.salary - :OLD.salary;     dbms_output.put_line('Old salary: '    :OLD.salary);     dbms_output.put_line('New salary: '    :NEW.salary);     dbms_output.put_line('Salary difference: '    sal_diff); END; /</pre>	
	<p><b>d) Ans.</b></p>	<p><b>Describe ACID properties of Transaction.</b></p> <p>A transaction is a single unit of execution. It can have different steps. Every transaction in the DBMS must follow the ACID properties. The ACID properties are:</p> <p><b>A-Atomicity</b>  <b>C-Consistency</b>  <b>I-Isolation</b>  <b>D-Durability</b></p> <p><b>Atomicity-</b>  This property states that every transaction should be treated as an atomic unit that is, either the entire transaction should be completed totally or it should not be done at all. It also states that under no condition should a transaction be partially completed.</p> <p><b>Consistency-</b>  The database must remain in consistent state after any transaction. The execution of a transaction should not result in inconsistency of the database.</p> <p><b>Isolation-</b>  In systems where more than one transaction execute simultaneously and in parallel, all transaction will be carried out and each transaction should feel that it is the only transaction happening. If <math>T_i, T_j</math> are two transactions, then <math>T_i</math> should feel that it is the only transaction happening while it is executing, either <math>T_j</math> should have completed execution or will execute once <math>T_i</math> completes.</p>	<p><b>4M</b></p> <p><i>1M for each property</i></p>



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<p><b>Durability-</b>          The database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction updates data in a database and commits, then the database will hold the modified data. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system starts.</p>	
	<p><b>e)</b>   <b>Ans.</b></p>	<p><b>Define database backup. Describe how database backup helps to avoid failures.</b></p> <p><b>Database backup:</b></p> <ul style="list-style-type: none"> <li>• Database Backup is storage of data that means the copy of the data.</li> <li>• It is a safeguard against unexpected data loss and application errors.</li> <li>• It protects the database against data loss.</li> <li>• If the original data is lost, then using the backup it can reconstructed</li> </ul> <p><b>The backups are divided into two types,</b></p> <ol style="list-style-type: none"> <li>1. Physical Backup</li> <li>2. Logical Backup</li> </ol> <p><b>1. Physical backups</b></p> <ul style="list-style-type: none"> <li>• Physical Backups are the backups of the physical files used in storing and recovering your database, such as data files, control files and archived redo logs, log files.</li> <li>• It is a copy of files storing database information to some other location, such as disk, some offline storage like magnetic tape.</li> <li>• Physical backups are the foundation of the recovery mechanism in the database.</li> <li>• Physical backup provides the minute details about the transaction and modification to the database.</li> </ul> <p><b>2. Logical backup</b></p> <ul style="list-style-type: none"> <li>• Logical Backup contains logical data which is extracted from a database.</li> <li>• It includes backup of logical data like views, procedures, functions, tables, etc.</li> <li>• It is a useful supplement to physical backups in many</li> </ul>	<p style="text-align: center;"><b>4M</b></p> <p style="text-align: center;"><i>1M for definition</i></p> <p style="text-align: center;"><i>3M for explaining how database backup helps to avoid failures.</i></p>



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

SUMMER – 2019 EXAMINATION  
MODEL ANSWER

Subject: Database Management

Subject Code: 22416

		<p>circumstances but not a sufficient protection against data loss without physical backups, because logical backup provides only structural information.</p> <p><b>Describe how database backup helps to avoid failures.</b> You may think that your data files can be prevented from any kind of loss by just replicating them to different locations in your disk which is normally called backup. Well, Regular backups are essential in the first place, but even they are subject to several failures that may include the loss or breaking of hard disks and other non-volatile storage devices.</p> <p>In case, any kind of failure strikes your SQL server databases, a proper plan and a way of recovery from that failure is needed, beforehand. A good backup can indeed be the best way to recover from most of the failures, especially when your data is too critical or important.</p> <p>In case, you lose a database or any of the table becomes corrupt, then you can simply reload your data from the backup. In addition, if you lose the whole server, then you may need to set up a new server and re-install the SQL Server backup software, before using any of your backups.</p> <p>One of the most effective ways to prevent any kind of data loss and to recover your original data in case of any failure is to store your entire SQL server database off-site. A secure off-shore backup can save you from many serious hassles in future.</p>	
5.	a)	<p><b>Attempt any <u>TWO</u> of the following:</b></p> <p>(i) Create the table as named 'student' with field as roll no, name, address, DOB and percent.</p> <p>(ii) Add a column 'city' to student table.</p> <p>(iii) Change the name of 'student' table to; 'student_info'.</p> <p>(iv) Remove/ delete the data or records from student info table.</p>	12 6M
	Ans.	<p>(i) Create the table as named 'student' with field as roll no, name, address, DOB and percent: create table student (</p>	



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<p>rollno number(5),  name char(20),  address varchar2(40),  DOB date,  percent number(5,2)  );</p> <p><b>(ii) Add a column ‘city’ to student table:</b>  alter table student add (city char(20));</p> <p><b>(iii) Change the name of ‘student’ table to; ‘student_info’:</b>  rename student to student_info;</p> <p><b>(iv) Remove/ delete the data or records from student info table:</b>  truncate table student_info ;  (OR)  delete from student_info;</p>	<p><i>Create table 3M</i></p> <p><i>Add 1M</i></p> <p><i>Change the name 1M</i></p> <p><i>Remove/delete data 1M</i></p>
	<p><b>b)</b> <b>Ans.</b></p>	<p><b>Describe Commit, Rollback and save point with example.</b>  <i>(Note: Any other example shall be considered)</i></p> <p><b>1) Commit:</b>  This command is used to end the transaction and also make its effect permanent to database. Commit deletes or removes the save points if any.  <b>Syntax:</b>  commit; (OR) commit work;</p> <p><b>2) Rollback</b>  A rollback command is used to undo the work done in current transaction.  <b>Syntax:</b>  Rollback;(or) Rollback work;</p> <p><b>Syntax to rollback to a particular save point:</b>  Rollback to savepoint&lt;savepoint_name&gt;;</p> <p><b>Example:</b> Rollback to save point SV1</p> <p><b>3) Save point:</b>  Save points define breakpoints for the transaction to have partial</p>	<p><b>6M</b></p> <p><i>Each description on 1M</i></p>



**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<p>rollback. Save points are treated as marker to divide lengthy transaction to smaller one.</p> <p><b>Syntax:</b></p> <pre>savepoint&lt;savepoint_name&gt;;</pre> <p><b>Example:</b> Savepoint SV1;</p> <p><b>Example showing usage of Commit, Rollback, Savepoint</b></p> <pre>SQL&gt;CREATE TABLE emp (   no NUMBER(3),   name VARCHAR(50),   code VARCHAR(12) ); SQL&gt;SAVEPOINT table_create; SQL&gt;insert into emp VALUES(1,'RAM', 'E101'); SQL&gt;SAVEPOINT insert_1; SQL&gt;insert into emp VALUES(2,'BEENA', 'E102'); SQL&gt;SAVEPOINT insert_2; SQL&gt;SELECT * FROM emp;</pre> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">NO</th> <th style="text-align: left;">NAME</th> <th style="text-align: left;">CODE</th> </tr> </thead> <tbody> <tr> <td colspan="3">-----</td> </tr> <tr> <td>1</td> <td>RAM</td> <td>E101</td> </tr> <tr> <td>2</td> <td>BEENA</td> <td>E102</td> </tr> </tbody> </table> <pre>SQL&gt;ROLLBACK TO insert_1; SQL&gt;SELECT * FROM emp;</pre> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">NO</th> <th style="text-align: left;">NAME</th> <th style="text-align: left;">CODE</th> </tr> </thead> <tbody> <tr> <td colspan="3">-----</td> </tr> <tr> <td>1</td> <td>RAM</td> <td>E101</td> </tr> </tbody> </table> <pre>SQL&gt;COMMIT;</pre>	NO	NAME	CODE	-----			1	RAM	E101	2	BEENA	E102	NO	NAME	CODE	-----			1	RAM	E101	<p><i>Example</i> <b>3M</b></p>
NO	NAME	CODE																						
-----																								
1	RAM	E101																						
2	BEENA	E102																						
NO	NAME	CODE																						
-----																								
1	RAM	E101																						
<b>c)</b>		<p><b>Give syntax for creating a view, Consider following schema-ACCOUNT (Account_No, Name, Account_Type, PAN_Number, Balance). Create a view on ACCOUNT having attributes (Account_No, Name, PAN_Number) where balance is less than 10,000.</b></p>	<b>6M</b>																					
<b>Ans.</b>		<p><b>Syntax for creating view:-</b></p>																						



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<p>Create [OR Replace][Force /Noforce] view          &lt;viewname&gt;[alias name ....]          as subquery          [with CHECK OPTION[CONSTRAINT]]          [with READ ONLY];</p> <p><b>Create view v1 as select Account_No, Name, PAN_Number from ACCOUNT where Balance&lt;10000;</b></p>	<p><i>Correct Syntax</i> <b>3M</b></p> <p><i>Correct Query</i> <b>3M</b></p>
<b>6.</b>	<b>a)</b>	<p><b>Attempt any <u>TWO</u> of the following:</b>  <b>Consider the following data base scheme student (roll_no, name, city, marks, result). Write queries for the following:</b>  <b>(i) Display all students having result as first class.</b>  <b>(ii) Update roll_no of each student by adding 18 to it.</b>  <b>(iii) Delete percent column from table.</b>  <b>(iv) Display student whose city is ‘Mumbai’</b></p>	<p><b>12</b> <b>6M</b></p>
	<b>Ans.</b>	<p><b>(i) Display all students having result as first class:</b>          select * from student where result= ‘first class’;</p> <p><b>(ii) Update roll_no of each student by adding 18 to it:</b>          update student set roll_no=roll_no+18;</p> <p><b>(iii) Delete percent column from table:</b>          alter table student drop column percent;</p> <p><b>(iv) Display student whose city is ‘Mumbai’:</b>          select * from student where city=’Mumbai’;</p>	<p><i>Each correct Query</i> <b>1½M</b></p>
	<b>b)</b> <b>Ans.</b>	<p><b>Describe different types of Indexes with examples.</b>  <b>Types of Index:</b></p> <p><b>1) Simple index (Single column):</b> An index created on single column of a table is called a Simple Index.  <b>Syntax:</b> Create index index_name on &lt;tablename&gt;&lt;column name&gt;;  <b>E.g.:</b> Create index idx on employee (empno);</p> <p><b>2) Unique indexes</b> are used not only for performance, but also for data integrity. A unique index does not allow any duplicate values to be inserted into the table.  <b>Syntax :</b>          Create unique index index_name on table_name(column_name);</p>	<p><b>6M</b></p> <p><i>Each index descripti on 1M</i></p>



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

		<p>E.g: Create unique index index_empno on emp(empno);</p> <p><b>3) Composite (concatenated):</b> Indexes that contain two or more columns from the same table which are useful for enforcing uniqueness in a table column where there's no single column that can uniquely identify a row.</p> <p><b>Syntax:</b> Create index index_name on &lt;tablename&gt;&lt;Column_name1, Column_name2&gt;;</p> <p><b>E.g.:</b> Create index idx on employee (ename, empno);</p>	<p><i>Each example 1M</i></p>										
c)	Ans.	<p><b>With suitable example write steps to create triggers and drop a trigger.</b>  <i>(Note: Any other example shall be considered)</i></p> <p><b>Creating a trigger:</b></p> <p><b>CREATE [OR REPLACE] TRIGGER trigger_name</b>  <b>BEFORE   AFTER</b>  <b>[INSERT, UPDATE, DELETE [COLUMN NAME..] ON</b>  <b>table_name</b>  <b>Referencing [ OLD AS OLD   NEW AS NEW ]</b>  <b>FOR EACH ROW   FOR EACH STATEMENT [ WHEN</b>  <b>Condition ]</b>  <b>DECLARE</b>              [declaration_section variable declarations; constant declarations; ]  <b>BEGIN</b>              [executable_section PL/SQL execute/subprogram body ]  <b>EXCEPTION</b>              [exception_section PL/SQL Exception block ]  <b>END;</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Keywords</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>CREATE [OR REPLACE]</b></td> <td>Creates the trigger. If already exist recreates the trigger definition without requiring the user to drop the trigger first.</td> </tr> <tr> <td><b>TRIGGERNAME</b></td> <td>Name of trigger to be created.</td> </tr> <tr> <td><b>BEFORE</b></td> <td>Oracle engine fires the trigger before executing triggering statement.</td> </tr> <tr> <td><b>AFTER</b></td> <td>Oracle engine fires the trigger after executing triggering statement.</td> </tr> </tbody> </table>	Keywords	Description	<b>CREATE [OR REPLACE]</b>	Creates the trigger. If already exist recreates the trigger definition without requiring the user to drop the trigger first.	<b>TRIGGERNAME</b>	Name of trigger to be created.	<b>BEFORE</b>	Oracle engine fires the trigger before executing triggering statement.	<b>AFTER</b>	Oracle engine fires the trigger after executing triggering statement.	<p><b>6M</b></p> <p><i>Trigger creation 3M</i></p> <p><i>Drop trigger 1M</i></p>
Keywords	Description												
<b>CREATE [OR REPLACE]</b>	Creates the trigger. If already exist recreates the trigger definition without requiring the user to drop the trigger first.												
<b>TRIGGERNAME</b>	Name of trigger to be created.												
<b>BEFORE</b>	Oracle engine fires the trigger before executing triggering statement.												
<b>AFTER</b>	Oracle engine fires the trigger after executing triggering statement.												



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

**SUMMER – 2019 EXAMINATION**  
**MODEL ANSWER**

**Subject: Database Management**

**Subject Code: 22416**

	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;"><b>DELETE</b></td> <td>Oracle engine fires the trigger whenever delete statement removes a row from the table.</td> </tr> <tr> <td><b>INSERT</b></td> <td>Oracle engine fires the trigger whenever insert statement add a row to the table.</td> </tr> <tr> <td><b>ON TABLENAME</b></td> <td>Define the table name to performing trigger event</td> </tr> <tr> <td><b>REFERENCING</b></td> <td>Give referencing to a old and new values of the data. :old means use existing row to perform event and :new means use executing new row to perform event.</td> </tr> </table> <p><b>Example:</b>          Let's create a table 'product_check' which we can use to store messages when triggers are fired.</p> <pre>CREATE TABLE product_check (Message varchar2(50), Current_Date date);</pre> <p><b>BEFORE UPDATE, Statement Level:</b> This trigger will insert a record into the table 'product_check' before a sql update statement is executed, at the statement level.</p> <pre>CREATE or REPLACE TRIGGER Before_Update_Stat_product BEFORE UPDATE ON product Begin INSERT INTO product_check Values('Before update, statement level',sysdate); END;</pre> <p>/</p> <p><b>Drop Trigger:</b>          Drop trigger Trigger_name;</p> <p><b>Example : Drop Trigger tr1;</b></p>	<b>DELETE</b>	Oracle engine fires the trigger whenever delete statement removes a row from the table.	<b>INSERT</b>	Oracle engine fires the trigger whenever insert statement add a row to the table.	<b>ON TABLENAME</b>	Define the table name to performing trigger event	<b>REFERENCING</b>	Give referencing to a old and new values of the data. :old means use existing row to perform event and :new means use executing new row to perform event.	<p><i>Any example for trigger creation 1M</i></p> <p><i>Trigger drop 1M</i></p>
<b>DELETE</b>	Oracle engine fires the trigger whenever delete statement removes a row from the table.									
<b>INSERT</b>	Oracle engine fires the trigger whenever insert statement add a row to the table.									
<b>ON TABLENAME</b>	Define the table name to performing trigger event									
<b>REFERENCING</b>	Give referencing to a old and new values of the data. :old means use existing row to perform event and :new means use executing new row to perform event.									