17332

16172

3 Hours / 100 Marks Seat No. | | | | | | | | |

Instructions : (1) All Questions are *compulsory.*

(2) Illustrate your answers with neat sketches wherever necessary.

(3) Figures to the right indicate full marks.

(4) Assume suitable data, if necessary.

Marks

1. (A) Attempt any SIX of the following : **12**

(a) Define Database. List any two properties of database.

(b) Define super key.

(c) List the notation used in relational algebra.

(d) Define Group by clause.

(e) What is SQL ? State its features.

(f) Define index.

(g) Define cursor.

(h) List advantages of Normalization

(B) Attempt any TWO of the following : **08**

(a) Describe the advantages of Distributed Database Management System.

(b) What is OUTER JOIN ? Explain in detail.

(c) Describe Exception handling. Explain with example.

2. Attempt any FOUR of the following : **16**

(a) Explain referential integrity constraints with example.

(b) Explain single value and multivalue attribute of E-R model.

(c) Explain string, data and time functions of SQL

(d) With the help of example, explain DROP VIEW command.

(e) Write a PL/SQL program to print numbers from 50 to 60 using for loop.

(f) List and explain types of DBMS users.

3. Attempt any FOUR of the following : **16**

(a) Explain having clause with example.

(b) Explain implicit and explicit cursor.

(c) Explain strong and weak entity set with example.

(d) Differentiate between view and index.

(e) Explain the concept of trigger.

(f) Describe function of database administrator.

4. Attempt any FOUR of the following : **16**

(a) Draw E-R diagram for customer and loan. Assume suitable attribute.

(b) Define following term with example

(i) Procedure

(ii) Function

(c) Describe commit and rollback with proper syntax.

(d) Consider the structure for book table as Book – master = {bookid, bookname, subcode-author, no_of copies, price} Write SQL queries for following :

(i) Display total no. of books for subject 'DBM'.

(ii) Get authorwise list of all books.

(iii) Display all books whose prices are between Rs. 200 and Rs. 500

(iv) Display all books with details whose name start with 'S'

(e) Describe unique index and composite index with example.

(f) List out query processing components and state their function.

5. Attempt any FOUR of the following : **16**

(a) Explain three tier architecture.

(b) Explain snapshot with example.

(c) Explain GOTO statement with example.

(d) Explain PL/SQL block structure.

(e) Explain candidate key and primary key.

(f) Explain concurrent execution of multiple transaction in detail.

6. Attempt any FOUR of the following : **16**

(a) Explain client – Server architecture.

(b) Explain Third normal form.

(c) Describe state of transaction with neat diagram.

(d) Draw E-R diagram for airline reservation system. Here a passenger can book ticket from personal for a flight on same date.

(e) Describe serializability.

(f) Explain locking strategy in detail.

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)**	**Attempt any SIX of the following:**	**12**
	(a) Ans.	**Define Database. List any two properties of database.** **Definition:** **Database** is defined as collection of related data. **Properties of database are:** 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.	**2M** ***Definition 1M*** ***Any 2 Properties ½ M each***
	(b) Ans.	**Define super key.** **Definition:** Super key of an entity set is a set of one or more attributes whose values uniquely determine each entity.	**2M** ***Definition 2M***
	(c) Ans.	**List the notation used in relational algebra.** 1. Selection(σ) 2. Projection (π) 3. Cartesian Product (X)	**2M** ***Any two notation***

		4. Natural Join(⋈) 5. Union(U) 6. Set Difference (–) 7. Rename (ρ)	***1 M each***
	(d) Ans.	**Define Group by clause.** The **SQL GROUP BY** 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** ***Description 2M***
	(e) Ans.	**What is SQL? State its features.** SQL is an acronym for Structured Query Language an is a standard relational query language used for creating and manipulating databases. **Features of SQL are:** • SQL allows the user to create, update, delete, and retrieve data from a database. • SQL is very simple and easy to learn. • SQL works with database programs like DB2, Oracle, MS Access, Sybase, MS SQL Sever etc.	**2M** ***Description 1M*** ***Any two features : ½ M each***
	(f) Ans.	**Define index.** **Index:** 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** ***Definition 2M***
	(g) Ans.	**Define cursor.** 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. **OR** A cursor is a temporary work area created in the system memory when a SQL statement is executed.	**2M** ***Definition 2M***
	(h) Ans.	**List advantages of Normalization.** **Advantages of the normalization.** • More efficient data structure. • Avoid redundant fields or columns. • More flexible data structure i.e. we should be able to add new rows and data values easily • Better understanding of data.	**2M** ***Any two advantages 1M each***

		• Ensures that distinct tables exist when necessary.	
1.	**(B)**	**Attempt any TWO of the following:**	**08**
	(a)	**Describe the advantages of Distributed Database Management System.**	**4M**
	Ans.	Following are the advantages of distributed databases over centralized databases. **1.Modular Development** − 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. **2.More Reliable** − 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. **3.Better Response** − If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response. **4.Lower Communication Cost** − In distributed database systems, if data is located locally where it is mostly used, then the communication costs for data manipulation can be minimized. **5.Local autonomy:** 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. **6. Expandability:** Expansion can be easily achieved by adding processing and storage power to the existing network.	***Any 4 advantages : 1M each***
	(b)	**What is OUTER JOIN? Explain in detail.**	**4M**
	Ans.	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. **Outer join are of three types:** **1. Left outer join** (also known as left join): this join returns all the rows from left table combine with the matching rows of the right	***Outer Join 2M***

		table. If we get no matching in the right table it returns NULL values. **Left Outer Join syntax :** SELECT column-name-list from table-name LEFT OUTER JOIN table-name2 on table-name1.column-name = table-name2.column-name; **2. Right outer join** (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. **Right Outer Join Syntax:**SELECT column-name-list from table-name1 RIGHT OUTER JOIN table-name2 on table-name1.column-name = table-name2.column-name; **3. The full outer join** keyword returns all records when there is a match in either left (table1) or right (table2) table records. **FULL OUTER JOIN Syntax:**SELECT column_name(s)from table1 full outer join table2 on table1.column_name = table2.column_name;	***Explanation of any 1 2M***
	(c) Ans.	**Describe Exception handling. Explain with example.** ***(Note: Any example can be considered)*** **Exception Handling:** 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. ***Syntax:*** **DECLARE** ; Declaration section …executable statement; **EXCEPTION** **WHEN ex_name1 THEN** :Error handling statements/user defined action to be carried out;	**4M** ***Description (syntax optional) 2M***

		END;
		DECLARE
		s_rollNostudents.rollNo%type := 10;
		s_namestudents.name%type;
		s_addressstudents.address%type;
		BEGIN
		SELECT rollNo, name, address FROM students WHERE rollNo = s_rollNo; dbms_output.put_line(s_rollNo || ' ' || s_name || ' ' || s_address);
		
		EXCEPTION
		WHEN no_data_found THEN
		dbms_output.put_line('No such student!');
		WHEN others THEN
		dbms_output.put_line('Error!');
		END;
			Example 2M

2.	**(a)** Ans.	**Attempt any FOUR of the following:** **Explain referential integrity constraints with example.** **(*Note: Any other example can be considered)*** **Referential integrity constraint:** • It is used to established a parent child relationship between two tables. • A value of foreign key is derived from the primary key. • 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. ***Syntax:*** Create table tablename (column datatype size references parenttablename (primary key attribute)….) ***Example:*** Create table product (EmpId number (4) references Emp (EmpId), PName varchar2(10)); **After table creation the foreign key is added as:** Alter table product add constraint fk_prod foreign key (EmpId) references Emp (EmpId);	**16** **4M** ***Description 2M*** ***Example2M***

	(b) Ans.	**Explain single value and multivalue attribute of E-R model.** **i) Single Valued attribute**: Attributes that can have single value at a particular instance of time are called single valued. ***Example***: age of a person is a single-valued attribute. ii)**Multi valued attributes**: A multi-valued attribute can have more than one value at one time. ***Example*:** phone_no of a person is a multi-valued attribute since a person can have more than one phone_no.	**4M** ***Description of each 2M***
	(c) Ans.	**Explain string, data and time functions of SQL.** SQL string functions are used primarily for string manipulation. Following are the string functions used in SQL. **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	**4M** ***Any 2 String functions 1M each***

Date Manipulation Functions:

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.

Any 2 date time functions 1M each

(d) **With the help of example, explain DROP VIEW command.** **4M**

Ans. DROP VIEW statement: Drops the specified view.

Explanation 2M

Syntax:
DROP VIEW view-Name**;**

Example:
DROP VIEW emp-view;

Example2M

Explanation:
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.

(e) **Write a PL/SQL program to print numbers from 50 to 60 using for loop.** **4M**

(Note: Any other Logic also considered)

Correct program 4M

Ans.
```
DECLARE
  x number :=50;
```

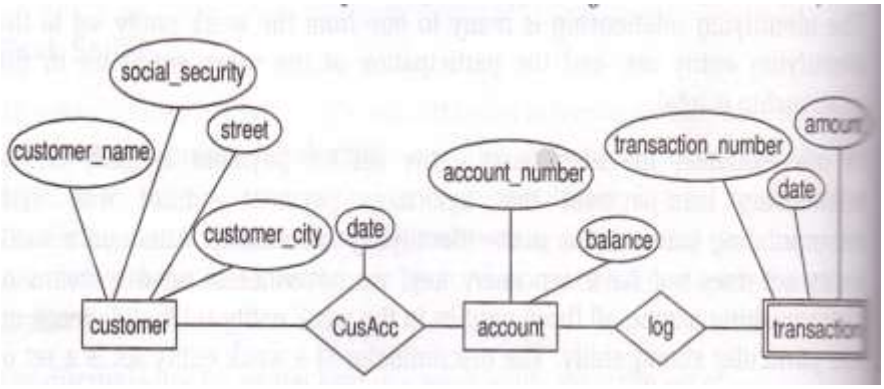
		BEGIN
		LOOP
		dbms_output.put_line(x);
		x := x +1;
		IF x >60 THEN
		exit;
		END IF;
		END LOOP;
		END;	

	(f) Ans.	**List and explain types of DBMS users.** **List of DBMS user:** a) Naive users b) Application programmers c) Sophisticated users d) Specialized users e) Database administrator **Explanation:** **a) Naive User:** • Naïve users are unsophisticated users • They are interact with the system through the application program. • They give data as input through application program or get output data which is generated by • application program. • Example: Bank cashier. **b) Application programmers:** • Application programmers are the users who write the program. • These programmers use programming tools to develop the program. • RAD technology is used to write the program. **c) Sophisticated users:** • Sophisticated users interact with the system by making the requests in the form of query language. • These queries are then submitted to the query processor. • Query processor converts the DML statements into lower	**4M** ***List 1M*** ***Explanation of any 3 1M each***

		level interactions which are understandable by storage manager. • Some sophisticated users can be analyst. **d) Specialized users:** • These users are not traditional. • They write some special application programs which are not regular applications. • Example: such types of applications are CAD, knowledge based and expert system. **e) Database administrator:** • Responsible for managing whole database system , create and maintains database. • Manages users who can access the database and manages integrity issue. • Manages performance of system as and when required.	
3.		**Attempt any FOUR of the following:**	**16**
	(a)	**Explain having clause with example.** ***(Note: Any other example considered)***	**4M**
	Ans.	Having clause is used to filter data based on the group functions. This is similar to WHERE condition but is used with group functions. Group functions cannot be used in WHERE Clause but can be used in HAVING clause. **Syntax** SELECT column1, aggregate_function(column2) FROM table1, table2 WHERE [conditions] GROUP BY group_expression HAVING [conditions] ORDER BY column1, column2 ***Example:*** SQL > SELECT ID, NAME, AGE, ADDRESS, MIN(SALARY) FROM CUSTOMERS GROUP BY ID HAVING MIN(SALARY)>=2000;	***Description 2M*** ***Example 2M***
	(b)	**Explain implicit and explicit cursor.**	**4M**
	Ans.	**Implicit Cursor:** Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the	

		statement.
		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.
		In PL/SQL, implicit cursor as has the attributes like %FOUND, %ISOPEN, %NOTFOUND, and %ROWCOUNT.
		
		Example of implicit cursor:
		Begin
		Update emp set salary= salary +500 where empno =&empno;
		If SQL%FOUND then
		Dbms_out.put_line(—Emp table modified‖);
		Else
		Dbms_out.put_line(—Emp table modified‖);
		End if;
		End;
		
		Explicit cursor:
		Explicit cursors are programmer defined cursors for gaining more control over the **context area**.
		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.
		
		Working with an explicit cursor involves four steps:
		Declaring the cursor for initializing in the memory
		Cursor cursor_name IS select_statement;
		Opening the cursor for allocating memory
		Open cursorname;
		Fetching the cursor for retrieving data
		Fetch cursorname INTO variable1,variable2…
		Closing the cursor to release allocated memory
		Close cursorname;
		
		Example of explicit cursor:
		Declare
		Cursor c1 is select empno, salary from emp Where deptno=10;
			Description or example 2M for each type

		ecode 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;	

	(c) Ans.	**Explain strong and weak entity set with example.** **Strong Entity Set:** An entity set that has sufficient attributes to form a primary key iscalled as strong entity set. *Example:* Cust_id is a primary key of Customer Entity so customer is strong entity set.. **Weak Entity set:** An entity set that does not have sufficient attribute to form a primary key is called as Weak Entity Set. *Example:* transaction is not having sufficient attribute to form a primary key so, transaction a weak entity set  	**4M** ***Each entity description 1M*** ***Example 1M each (diagram optional)***

MODEL ANSWER

SUMMER - 2017 EXAMINATION

Subject: Relational Database Management System **Subject Code:** **17332**

(d) Ans.

Differentiate between view and index. — **4M**

Parameter	View	Index
Definition	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.
Function	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
Syntax	CREATE VIEW view_name AS SELECT column1, column2..... FROM table_name WHERE [condition];	Create index index_name on Table_name (coloumn_name);
Example	CREATE VIEW CUSTOMERS_VIEW AS SELECT name, age FROM CUSTOMERS;	create index s_index on Student(roll_no);

Any 2 relevant points : 2M Each

(e) Ans.

Explain the concept of trigger. — **4M**

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.

Syntax for Creating a Trigger:

```
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
```

Example:

```
CREATE OR REPLACE TRIGGER trg1
```

Description 4M (Example Optional)

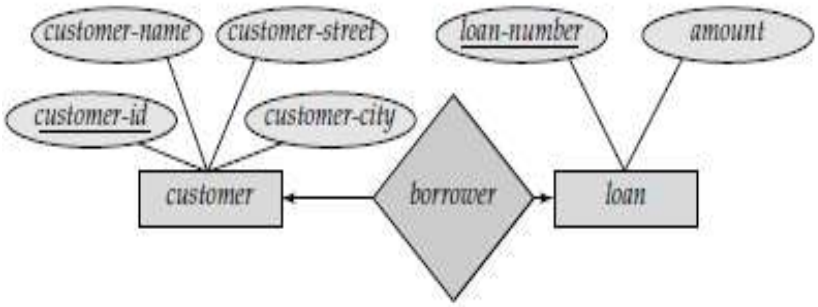
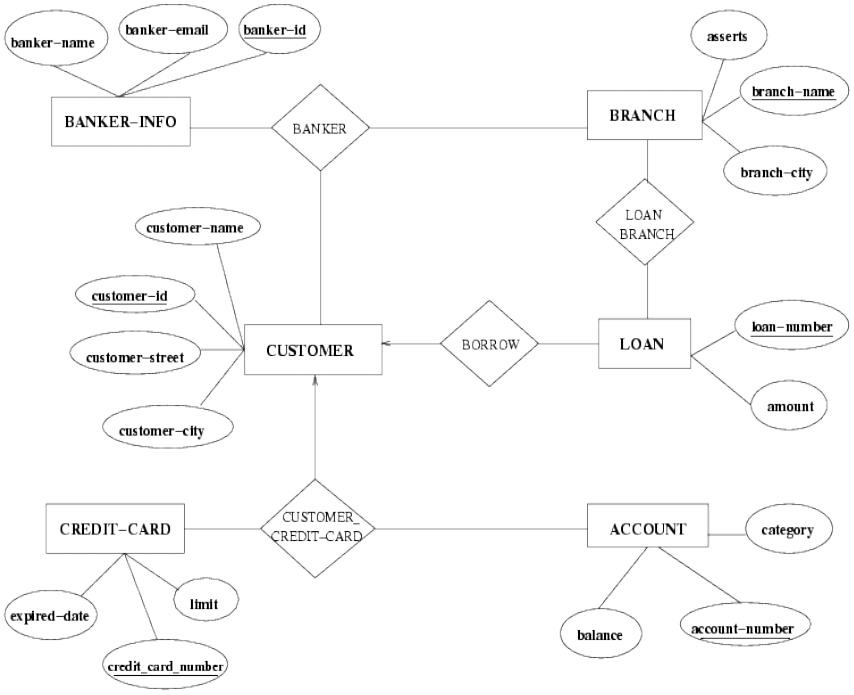
		BEFORE INSERT ON EMP
		FOR EACH ROW
		BEGIN
		IF :new.sal<=0 THEN
		Rasie_application_error('Salary should be greater than 0');
		END IF;
		END;	

	(f) Ans.	**Describe function of database administrator.** **Function of database administrator** **1.Schema Definition** 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. **2. Storage structure and access method definition** The DBA creates appropriate storage structures and access methods by writing a set of definitions which is translated by data storage and DDL compiler. **3. Schema and physical organization modification** DBA writes set of definitions to modify the database schema or description of physical storage organization. **4. Granting authorization for data access** 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. **5. Routine Maintenance** 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.	**4M** ***Any 4 functions :1M each***

MODEL ANSWER

SUMMER - 2017 EXAMINATION

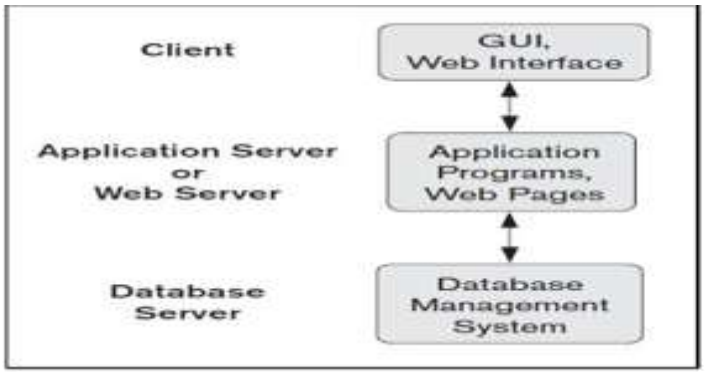
Subject: Relational Database Management System **Subject Code:** **17332**

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

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

	(c) Ans.	**Describe commit and rollback with proper syntax.** **Commit:** The COMMIT command is used to save changes invoked by a transaction to the database. The COMMIT command saves all transactions to the database since the last COMMIT or ROLLBACK command. The syntax: SQL> COMMIT; Or COMMIT WORK; **Rollback:** The ROLLBACK command is used to undo transactions that have not already been saved to the database. The ROLLBACK command can only be used to undo transactions since the last COMMIT or ROLLBACK command was issued. We can either rollback the entire transaction or till a particular save point transaction can be rolled back. **The syntax for ROLLBACK is:** **ROLLBACK TO SAVEPOINT_NAME;** **OR** **ROLLBACK;** **OR** **ROLLBACK WORK;**	**4M** ***For each explanation :1M*** ***Syntax 1M each***
	(d) Ans.	**Consider the structure for book table as Book-master = {bookid, bookname, subcode-author, no_of copies, price} Write SQL queries for following:** **(i) Display total no. of books for subject 'DBM'** **(ii) Get authorwise list of all books.** **(iii) Display all books whose prices are between Rs.200 and Rs.500** **(iv) Display all books with details whose name start with 'S'** ***(Note: Queries with any other correct logic shall be considered)*** (i)select sum(no_of Copies) from Book_master where subcode= ' DBM'; (ii) Select SUM(no_of Copies),author from Book_master group by (author); (iii) Select bookname from book_master where price>=200 and price	**4M** ***Each correct query 1M***

		<=500; (iv) select * from book_master where bookname like 'S%';	
	(e) Ans.	**Describe unique index and composite index with example.** **Unique Index:** Unique indexes are used not only for performance, but also for data integrity. An unique index does not allow any duplicate values to be inserted into the table. The basic syntax is as follows: **CREATE UNIQUE INDEX index_name on table_name (column_name);** ***Example:*** CREATE UNIQUE INDEX ename_idx on emp (ename); **Composite Index:** A composite index is an index created on two or more columns of a table. The basic syntax is as follows: CREATE INDEX index_name on table_name (column1, column2); ***Example:*** CREATE INDEX en_idx on emp (ename,job);	**4M** ***Explanation of each 1 M*** ***Example of each 1M***
	(f) Ans.	**List out query processing components and state their function.** **Components of query processing:** **1)DDL interpreter** **2) DML compiler** **3) Embedded DML PreCompiler** **4) Query Evaluation Engine** **The query processor components include:** 1. **DDL Interpreter**: It interprets DDL statements and records them in a set of tables containing metadata or data dictionary. 2**. DML Compiler**: It translates DML statements of high level query language into low level instructions that query evaluation engine understands. 3. **Embedded DML PreCompiler**: It converts DML statements	**4M** ***List 1M*** ***Description of any 3 1M each***

		embedded in application program to normal procedural calls in host language. The precompiler must interact with the DML compiler to generate the appropriate code. 4. **Query Evaluation Engine**: It executes low level instructions generated by DML compiler and DDL	
5.	**(a)** Ans.	**Attempt any FOUR of the following:** **Explain three tier architecture.** 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. 	**16** **4M** ***Explanation 4M*** ***(Diagram optional)***
	(b) Ans.	**Explain snapshot with example.** **Snapshot:** 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 **It has two types:** Simple snapshot and complex snapshot. Simple snapshot related to single table and complex snapshot related to joined	**4M** ***Snapshot explanation 2M***

		tables. ***Example :*** **Operations on snapshot:** **i) Creating Snapshot:** Create snapshot command is used to create the snapshot. *Syntax:-* CREATE SNAPSHOT [schema.] <snapshot name>AS subquery; *Example:-* Create snapshot emp_snapas select * from emp where deptno=6; **ii) Altering snapshot:** Snapshot can be altered by using ALTER SNAPSHOT command. The only parts ofa 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. *Syntax:-* ALTER SNAPSHOT <snapshotname> [[PCTFREE <integer>] [PCTUSED <integer>] [REFRESH [FAST/COMPLETE/FORCE]]; *Example:-* To change the automatic refresh mode for the emp_data snapshot to fast: ALTER SNAPSHOT emp_data REFRESH FAST; **iii) Dropping a snapshot** To remove the snapshot DROP SNAPSHOT Command .When snapshot is dropped which a snapshot log had associated with it, only the rows required for maintainingthat snapshot are dropped. *Syntax:-* Drop snapshot <snapshot name>; *Example:-* Drop snapshot emp_snap;	***Any one operations example :2M***

	(c) Ans.	**Explain GOTO statement with example.**
		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.
		Syntax :
		GOTO label;
		..
		..
		<< label >>
		statement;
		
		Example :
		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';
		<<print_now>>
		DBMS_OUTPUT.PUT_LINE(TO_CHAR(n) || p);
		END;
			4M
			
			Explanation 2M
			
			Example 2M (any other example considered)

	(d) Ans.	**Explain PL/SQL block structure.** **Block structure of PL/SQL:** Declare *Declaration of memory variables* BEGIN (Mandatory) *SQL executable statements* Exception	**4M** ***Correct block structure 3M***

		Handling errors END; (Mandatory) 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	***Explanation 1M***
	(e) Ans.	**Explain candidate key and primary key.** **Candidate key:** 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. **Primary key:** 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.	**4M** ***2M each***
	(f) Ans.	**Explain concurrent execution of multiple transaction in detail.** A transaction is a collection of operations that perform modifications in database application. A set of transactions can be executed by some sequence which is called as a schedule. There are two types of schedules as serial executions and concurrent execution. 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. Concurrent execution should always preserve consistency. This means that even if the transactions are interleaved, there should not be any damage to data. *Example :* 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 :	**4M** ***Explanation 4M***

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

Initially A=100, B=150 therefore A+B=250.

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

6. **Attempt any FOUR of the following:** **16**

(a) **Explain client-Server architecture.** **4M**

Ans. Client server is a system in which all the data & information is stored on the server & all application programs are stored on the client. In this client server relationship, several clients can share the same server. ***Explanation 4M***

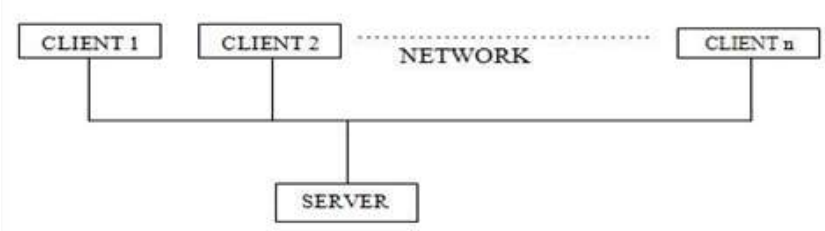


Diagram Optional

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

clients.
There are different types of client/server architecture such as
- Two tier architecture,
- Three tier architecture.

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

(b) Ans.

Explain Third normal form.

- **3NF** (**Third normal form**)
  1) After removing all transitive dependencies and making separate relations, relations get into 3NF.
  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.

4M

Example

Student_Detail Table :

Student_id	Student_name	DOB	Street	city	State	Zip

Explanation 3M

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 **transitive dependency**. Hence to apply **3NF**, we need to move the street, city and state to new table, with **Zip** as primary key.

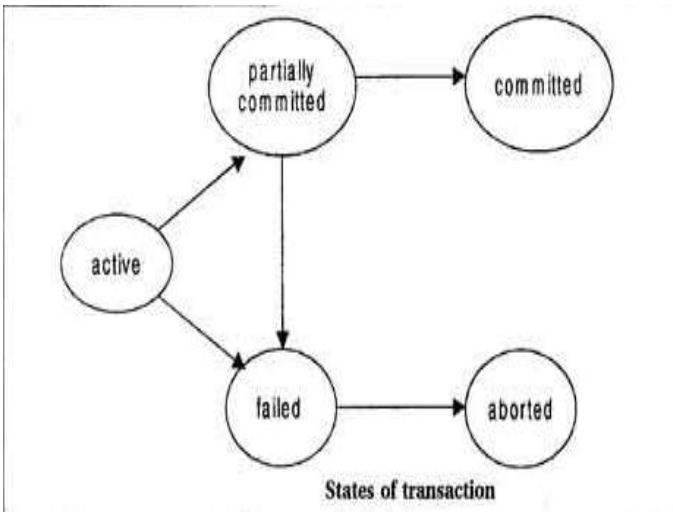
Example 1M

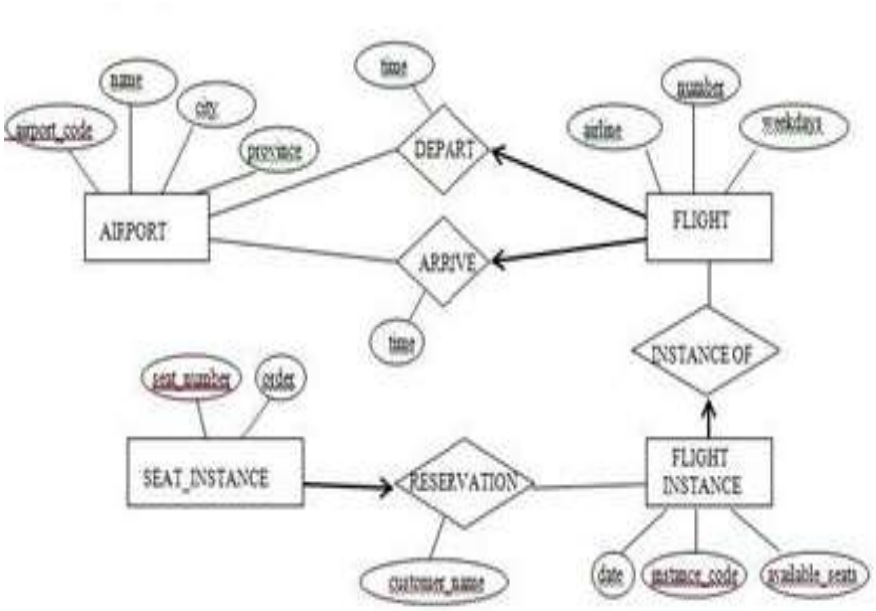
New Student_Detail Table :

Student_id	Student_name	DOB	Zip

Address Table :

Zip	Street	city	State

	(c) Ans.	**Describe state of transaction with neat diagram.**  States of transaction A transaction must be in one of the following states: 1. **Active**: the initial state, the transaction stays in this state while it is executing. 2. **Partially committed**: after the final statement has been executed. 3. **Failed**: when the normal execution can no longer proceed. 4. **Aborted**: after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction. 5. **Committed**: after successful completion.	**4M** ***Diagram 2M*** ***Explanation 2M***

	(d) Ans.	**Draw E-R diagram for airline reservation system. Here a passenger can book ticket from personal for a flight on same date.**  	**4M** ***Correct use of symbols 2M*** ***Correct entities and relation ships 2M***
	(e) Ans.	**Describe serializability.** 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. Serializability ensures consistency of database. *Example :* 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 T1 | T2 Read(A); ……A=100 | A:=A-50; | Write(A); …..A=50 |	**4M** ***Description with suitable example 4M***

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

	(f) Ans.	**Explain locking strategy in detail.** **Implicit Locking:** Implicit locks are generally placed by the DBMS automatically. **Explicit Locking:** 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 **1) The Select.....For Update statement** It is used for acquiring exclusive row level locks in anticipation of performing updates on records. **2) Using lock table statement:** To manually override Oracle's default locking strategy by creating a data lock in a specific mode. ***Syntax:*** **LOCK TABLE <TableName> [, <TableName>]…** **IN { ROW SHARE| ROW EXCLUSIVE|SHARE UPDATE|SHARE|SHARE ROW EXCLUSIVE|EXCLUSIVE} [NOWAIT]**	**4M** ***Explanation 4M***