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.

1. Attempt any ten: [Marks 20]

a) What do you mean by data model?

[Definition 2 marks]

Data model is the collection of conceptual tools for describing data, data schema, data relationship, consistency constraints.

b) Define (i) Primary key (ii) Candidate key.

[Primary Key 1 Mark, Candidate Key 1 Mark]

Primary key: Within a relation there is always one attribute which has values that are unique in a relation and thus can be used to uniquely identify tuple of that relation. Such a unique identifier is called the primary key. E.g. In the Student relation RNO is the primary key.

Candidate Key: Sometimes in a relation there are more than one attributes are having the unique identification property. Those attributes are known as candidate keys.E.g. Consider Student (RNO, NAME, PER, BRANCH),if RNO and NAME are unique then both are known as Candidate keys.

c) What is meant by constraint?

[Definition 2 Marks]

{Note: Any other relevant definition should be given marks}

Constraints are used to ensure accuracy and consistency of data in a relational database.

d) Define data independence and list its types.

[Definition 1Mark, Types 1 Mark]

The ability to modify a schema definition in one level without affecting a schema definition in next higher level is called data independence.

There are two types of data independence 1. Physical data independence 2.Logical data independence.

e) Write features of SQL.

[Any four features - 1/2Mark each]

1. SQL is an easy-to-understand language having English-like structure.
2. It is a comprehensive tool for creating and managing data.
3. It is not case sensitive.
4. It supports client server architecture.
5. It is a non procedural language.
6. It is an important tool for communicating with the DBMS and supports DBMS statements.

f) What is meant by lock based protocol?

[Explanation 2 Marks]

A lock is a mechanism to control concurrent access to a data item. A locking protocol is a set of rules followed by all transactions while requesting and releasing locks.

g) What is deadlock?

[Definition 2 Marks]

A situation in which all transactions are waiting for each other to release the data item and cannot get executed is known as deadlock.

h) What is meant by concurrent execution?

[Definition 2 Marks]

When two or more transactions are executing simultaneously, the execution is known as concurrent execution. OR

In concurrent execution, the database management system controls the execution of two or more transactions in parallel.

i) List components of DBMS system.

[Any 4 components 1/2 Mark each]

Different components of DBMS are:

1. Query Processor (DDL interpreter, DML compiler, Embedded DML Pre-Compiler, Query Evaluation Engine)

2. Storage Manager (Authorization and Integrity Manager, Transaction Manager, File Manager, Buffer Manager)

3. Disk Storage(. Data Files, . Data Dictionary, Statistical data, indices)

j) List advantages of DBMS.

[Any 2 advantages, 1 Mark for each advantage]

1. Reduction in Redundancy: Duplication of records is reduced.
2. Avoiding Inconsistency: As the redundancy is reduced inconsistency is avoided.
3. Maintaining Integrity: Accuracy is maintained.
4. Sharing of data: Sharing of data is possible.
5. Enforcement of Security: Security can be enforced.
6. Transaction support.

k) What is Relational Algebra?

[Explanation 2 Marks]

Relational Algebra is a procedural query language. It consists of a set of operations that take one or two relations as input and produces a new relation as their result.

l) List different string functions in SQL.

[Any 4 string functions. ½ mark each]

String functions: inticap,lower,upper,ltrim,rtrim,translate,replace,substr,lpad,rpad,length,concatenation.

2. Attempt any four: [Marks 16]

a) What are the components of DBMS? Explain in brief.

[Listing components 1 Mark, Explanation 3 Marks]

Components of DBMS are classified in three categories:

1. Query Processor:

- DML Compiler : It translates DML statements of High level language into low level instructions that query evaluation engine understands.
- Embedded DML Pre-Compiler: It converts DML statements embedded in application program to normal procedural calls in host language.
- DDL Interpreter: It interprets DDL statements and records them in a set of tables containing metadata.
- Query Evaluation Engine: It executes low level instructions generated by DML compiler and DDL interpreter.

2. Storage Manager Components:

- Authorization and Integrity Manager: It tests for integrity constraints and authority of the user.

- Transaction Manager: It ensures that the database remains in consistent state despite the system failures and that concurrent transaction execution proceeds without conflicting.
- File Manager: It manages the allocation of space on disk storage& data structures used to represent information stored on disk.
- Buffer Manager: It is responsible for fetching data from disk storage into main memory and deciding what data to cache in memory.

3. Disk Storage:

- Data Files: It stores the database.
- Data Dictionary: It stores metadata about the structure of the database.
- Indices: Provide fast access to data items that hold particular values.
- Statistical Data: It stores statistical information about the data in the database. This information is used by query processor to select efficient ways to execute query.

b) Explain BCNF.

[Definition 1 Mark. Explanation with any suitable example 3 Marks]

Definition: A relation R is in Boyce-Codd normal form (BCNF) if and only if every determinant is a candidate key. OR

BCNF: A relation R is in BCNF if whenever on nontrivial functional dependency X→A holds in R, then X is a superkey of R.

Consider the relation SUPPLIER (SNO, SNAME, PH_NO, CITY) having SNO and SNAME unique. In this there are 2 determinants SNO, SNAME as PH_NO and CITY dependence upon them and both are candidate keys. So this is in BCNF.

c) Explain (i) Data definition language (ii) Data manipulation language.

[DDL-2 Marks,DML-2 Marks]

Data Definition Language (DDL) commands provide means for defining relation schema, deleting relations, creating indices, views and modifying relation schemas. E.g.

CREATE TABLE/VIEW/INDEX to create a table/view/index in the database

DROP TABLE to remove a table from the database

ALTER TABLE to add, modify or remove columns from a table in the database

RENAME to change the name of table.

2) Data Manipulation Language (DML) commands manipulates the data i.e. they perform the operations of insertion, deletion, updating and selection of data.

SELECT to select rows of data from a table

INSERT to insert rows of data into a table

UPDATE to change rows of data in a table

DELETE to remove rows of data from a table

d) Explain following operations in relational algebra with example (1)Selection (2)Projection

[Select operation 2 Marks, Project operation 2Marks]

Relational Algebra is a procedural query language. It consists of a set of operations that take one or two relations as input and produces a new relation as their result.

(1) SELECTION(σ)

SELECT is used to obtain a subset of the tuples of a relation that satisfy a *select condition*. OR

The select operation selects tuples that satisfy a given predicate.

Comparison can be done by using =,<,>. Several predicates can be combined together by using logical and(^) and or(∨) operators.

For example, 1.To find all employees having salary more than 10000

$$\sigma_{\text{salary} > 10000}(\text{employee})$$

2.To find students having dept as 'CM' and percentage more than 80.

$$\sigma_{\text{dept='CM'} \wedge \text{per}>80}(\text{student})$$

(2) PROJECTION (π)

Projection is a unary operation which is used to select a subset of the attributes of a relation by specifying the names of the required attributes.

For example, to get a list of empno and name:

$$\pi_{\text{name, empno}}(\text{employee})$$

e) What is Normalization? What are different normal forms?

[Normalization-1 Mark, Types-3 Marks]

Definition: In relational database design, the process of organizing data to minimize redundancy is known as Normalization. OR

A relation is said to be in a particular normal form if it satisfy certain specified set of conditions.

Different Normal Forms are: First Normal Form(1NF),Second Normal Form(2NF),Third Normal Form(3NF),Boyce Codd Normal Form(BCNF),Fourth Normal Form(4NF),Fifth Normal Form(5NF).

1NF: A relation is said to be in 1NF if the values in the domain of each attribute of the relation are atomic i.e. only one value associated with each attribute.

2NF: A relation R is in 2NF if and only if it is in 1NFand every non-key attribute is fully functionally dependent on the primary key.

3NF: A relation is in 3NF if and only if it is in 2NF and every non key attribute is non transitively dependent on the primary key.

BCNF: The relation R is in BCNF if and only if all the determinants are candidate keys.

f) Explain following commands with syntax and example (i)ALTER (ii)UPDATE

[Alter command 2 Marks, Update command 2 Marks]

- Alter table command is used to add new column to the table or to modify the column width and data type of the column or to drop any particular column of the table.
  - **Syntax:** alter table <table name> add(column _name datatype(size));
    e.g. 1.alter table emp (add dob date);
  - alter table <table _name>modify (column_name datatype(size));
    e.g. alter table emp modify (ename varchar2(30));
- Update: Update command is used to modify the existing data of any relation.
  - **Syntax:** Update table_name set column_name1=expression1>;
    e.g. update emp set bonus=2000;
  - Update table_name set column_name1=expression1> where <search condition>;
    e.g. update emp set bonus=1000 where designation= 'Programmer';

3. Attempt any four: [Marks 16]

a) Explain in brief relational calculus.

[Explanation 3 Marks, Example 1 Mark]

Relational calculus refers two categories: 1)Tuple relational calculus and 2) Domain relational calculus which are the part of relational model for database and that provide declarative ways to specify database query. It's a non procedural way of expressing queries.

1. **Tuple relational calculus**:-It is a non procedural calculus. It describes information without giving a specific procedure for obtaining that information. A query in tuple calculus is expressed as {t|p(t)} i.e. the set of all tuples (t) such that predicate (P[condition]) is true for 't'.

We use 't[a]' to denote the value of tuple on attribute 'a' & we use 'tϵ r' to denote that tuple 't' is in relation 'r'.There are different symbols with specific meaning which can be used to write tuple calculus expression;-

1. ε- belong to
2. ∃-There exits
3. ∀ – for all
4. ¬ – not
5. => implies
6. ^ -and
7. ∨- or

Example.

Find records of employees where salary is more than 20000.

{t| s ε emp(s[salary]>20000)}

Domain relational calculus:- It represents domain variables instead of tuple .The standard format & symbols are similar to tuple calculus expression .Formal definition of domain relational calculus:

An Expression in the form of domain relational calculus is of the form:

{<x1,x2,x3,….xn>|P(x1,x2,…..xn)}

Where x1,x2,x3…xn represents domain variable and P represents predicate example:

Eg : Find names & salaries of all employers belonging to department no 10.

{<n,s>| <n,s> ε emp (d=10)}

Where n-domain of name.

s-domain of salary

d-domain of department.

b) Give syntax and example of VIEWS.

[Explanation 2 Marks, Syntax 1 Mark, Example 1Mark]

A view is a logical extract of a physical relation i.e. it is derived from any base relation. Hence, it is called as a derived relation. A view can hide data that user does not need to see. It helps to enhance security. A user may deny direct access to relation but may be allowed to access part of relation with the help of views.

Views do not exit physically .Views are virtual relations mainly used for security purpose, and can be provided on request by a particular user. A view can be created by joining one or more tables. When you update records in view, it updates the records is the underlying table that make up the view.

Syntax for creating view:-

Create view<viewname> as select <query>

Example:-

- Create view V1 as select empno,empname from emp where salary >20000;
- Select * from V1;

Here'V1' is the view containing data for empno & empname from emp table where salary is more than 20000.

c) Explain referential integrity constraints.

[Explanation 3 Marks, Example 1 Mark]

{Note: Any one syntax of declaration of foreign key and an example can be considered.}

It represents relationship between tables. A foreign key is a column or group of column whose values are derived from some other table. A foreign key must have a corresponding primary key value in some other table to have some value. It rejects insert or update if corresponding value is not present in parent table. Foreign key rejects delete if it invalidates reference.

There is additional clause in foreign key

On delete cascade- When this clause is applied to a foreign key ,if a record from main table is deleted, automatically all records from other table corresponding to foreign key values gets deleted.

Syntax to declare a foreign key:-

a) Column level-

Create table <table name> (col1 data type(size),col2 data type(size) constraint<constraint name> references<parent table name>(parent table-column name),col3 data type(size)…..)[on delete/update cascade];

b) Table level-

Create table <table name > (col1 data type(size),Col2 data type(size),….) constraint foreign key (column name) reference<parent table name> (parent table-column name)[on delete/update cascade];

Example-

Create table emp(empno number(5),deptno number(3),..) constraint f_k(deptno) references dept(deptno);

It can be done later by 'Alter'as

Alter table emp add constraint f_k foreign key(deptno) references dept(deptno);

d) Explain following terms with suitable example (i) simple attributes (ii) composite attributes

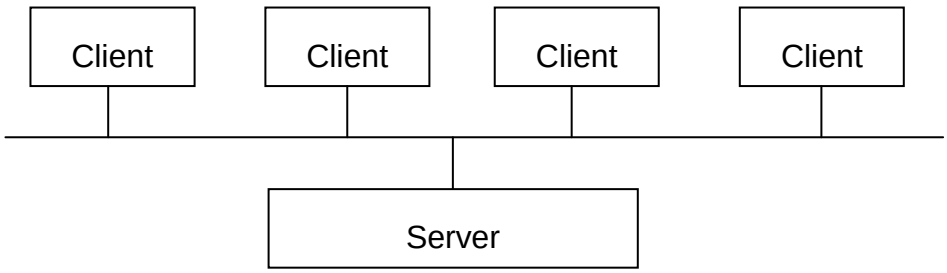
[Each type 2 Marks]

Simple attribute- The attributes that cannot be divided into smaller parts, are called as simple attributes. E.g. rollno, age, deptno etc.

Composite attributes- The attributes which can be divides further into smaller parts, which represent more basic attributes with independent meaning are called as composite attributes .E.g. date_of_birth can be divided into dd, mm & yy. So date_of_birth is a composite attribute.

e) What is client server architecture? Draw Diagram.

[Explanation 2 Marks, Diagram 2 Marks]



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, 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.

f) What is indexing? Explain any one type of indexing.

[Indexing 2 Marks, One type of index 2 Marks]

Index structures are used to speed up the retrieval of records is response to certain condition. The index structure typically provides alternative ways of accessing records, without affecting the physical placement of the records on disk. They enable efficient access to records based on some indexing field that are used to construct the index. Basically any field can be used to create an index and multiple indices on different fields can be constructed on the same file .A variety of indices are possible. To find

record or records in the file based on certain selection criteria on an indexing field one has to initially access the index which points to one or more blocks in the file where required records are located, there are basically two types of indices as single level; indices and multiple indices.

Single level indexing:-

{Note-Any one type out of five as 1.Primary 2.Secondary 3.Clustering index 4.Dense index 5. Sparse index can be explained. As an example primary index is explained here}

Primary index- It is an ordered file whose records are of fixed length with two fields. First field is of the same data type as the ordering field called as primary key & second field is a pointer to the disk block. There is one index entry is the index file for each block in the data file. Each index entry has the value of the primary key field for 1st record in the block and a pointer to that bock on its two field value.

4. Attempt any four: [Marks 16]

a) Explain network model in brief.

[Explanation 3 Marks, Diagram 1 Mark]

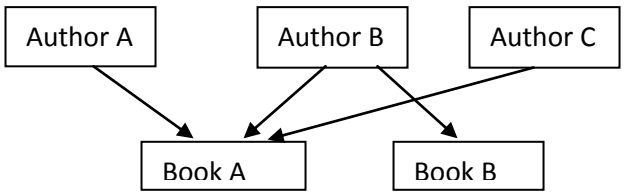
{Note: Any other example showing features of network model can be considered.}

Data modeling is a collection of conceptual tools for describing data, relationships, constraints etc.

Data models can be categorized into three different groups: 1) object based 2) record based & 3) physical model.

Record based logical data models are used in describing the data at the conceptual and view level. There are 3 widely used record based models, out of which Network model is one type.

Network model: In this model, the records in the database are organized as a collection of nodes and links. Unlike hierarchical model, in network model, a node may have more than one parent node.



Example representing Network model.: It shows that there can be more than one authors for a book same as in network model a sub node may have more than one parent node.

b) Explain and list set of operators in SQL with example.

[Each operator 1 Mark]

{Note: Any other operators like Arithmetic, Comparison, Logical operators should be consider}

Set operators combine the results of two component queries into a single result. Queries containing set operators are called as compound queries. Set operators in SQL are represented with following special keywords as : Union, Union all, intersection & minus.

Consider data from two tables emp1 and emp2 as

emp1	emp2
ename	ename
abc	pqr
xyz	xyz
lmn	

1) **Union** : The Union of 2 or more sets contains all elements, which are present in either or both. Union works as or.

   Eg select ename from emp1 union select ename from emp2;

   The output considering above data is :

   Ename
   abc
   xyz
   lmn
   pqr

2) **Union all** : The Union of 2 or more sets contains all elements, which are present in both, including duplicates.

   Eg select ename from emp1 union all select ename from emp2;

   The output considering above data is :

   Ename
   abc
   xyz
   lmn
   pqr
   xyz

3) **Intersection:** The intersection of 2 sets includes elements which are present in both.

Eg select ename from emp1 intersection select ename from emp2;

The output considering above data is:

Ename

Xyz

4) **Minus:** The minus of 2 sets includes elements from set1 minus elements of set2.

Eg select ename from emp1 minus select ename from emp2;

The output considering above data is:

Ename

abc

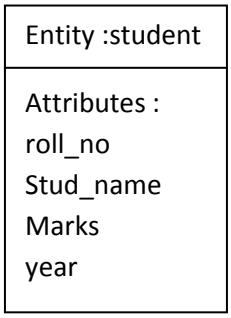
lmn

c) Explain following terms with suitable example (i) Entity (ii) Entity set

[For each 2 Marks]

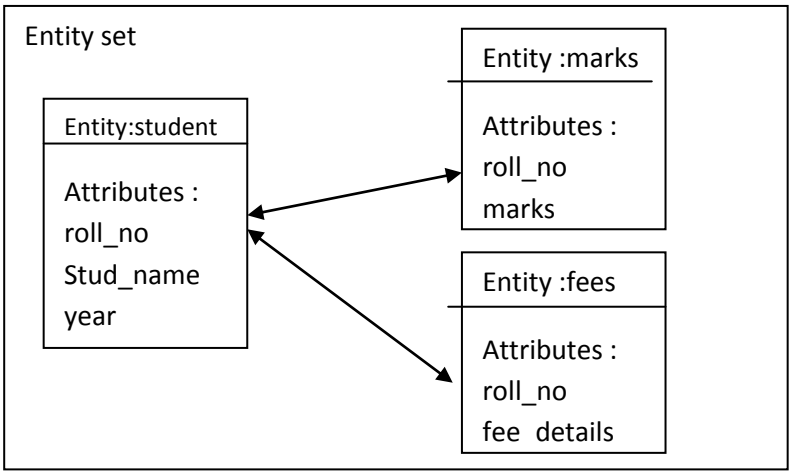
1) Entity: Entity is an object which can be described by specific set of attributes. For example : 'Student' is an entity which can be described by specific number of attributes such as roll_no, stud_name, marks,year etc

Example:



2) Entity set: A collection of all entities of a particular type is called as entity set. For example for keeping the data of students, one can use entities like student_details, fee_details, marks_ details which are separate entities but collectively keeping data of student. So collectively they form entity set.

Example:



d) With suitable example list and explain types of cursor.

[For each type 2 Marks]

{Note: any other examples showing use of implicit and explicit cursor can be considered.}

Cursor: A work area used for internal processing in order to execute SQL statement is called as a cursor.

Cursors are classified as implicit cursor and explicit cursor.

Implict cursor: If database engine opens a cursor for internal processing, it is called as implicit cursor.

Example of implicit cursor:

```
Begin
Update emp set salary= salary +500 where empno =&empno;     ----->[ implicit cursor created]
 If SQL%FOUND then    - -->[ checks whether data is fetched successfully in implicit cursor or not]
    Dbms_out.put_line("Emp table modified");
Else
     Dbms_out.put_line("Emp table modified");
End if;
End;
```

Explicit cursor: A user can open a cursor for processing data as required. Such user defined cursors are known as explicit cursors.

Example of explicit cursor:

```
Declare
Cursor c1 is select empno, salary from emp  Where deptno=10;    → explicit cursor c1 created.
 Ecode emp.empno%Type;
Sal emp.salary%Type;
Begin
        Open c1;                          → explicit cursor c1 opened
        If c1%ISOPEN then
              Loop
        Fetch c1 into ecode,sal;          → values from cursor c1 fetched
        If c1% NOTFOUND then
        Exit;
        End if;
                Update emp set salary = salary+500;
        End Loop;
                Close c1;                                  → explicit cursor c1 closed.
         Else
        Dbms_out.put_line("unable to open");
         End if;
```

End;

e) What is cardinality? How its mapping is done?

[Cardinality 1 Mark, Mapping 3 Marks]

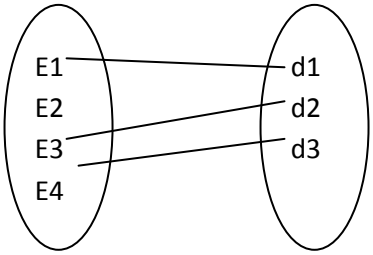
{Note: any other examples showing use of implicit and explicit cursor can be considered.}

Cardinality: Total number of records from a relation is called as cardinality of a relation.

For eg : If there are three records in relation student then cardinality of relation student is 3.

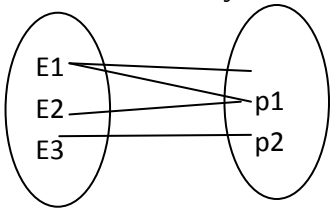
Mapping cardinality:

1) **One to one** : This kind of mapping exists when an instance of one entity associates with only one instance of another entity in an entity set.



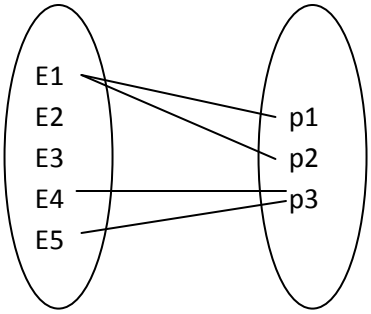
Emlpyee manages a department (1:1 mapping)

2) **One to many**: This kind of mapping exists when an instance of one entity may associate with one or many instances of another entity in an entity set.



Employee works on different projects. (1: M mapping)

3) **Many to many** : This kind of mapping exists when one or more instances of one entity may associate with one or many instances of another entity in an entity set.



Many employees works on many projects (M:N mapping)

f) Explain following commands with syntax and example (i) CREATE TABLE (ii) INSERT.

[For each 2 Marks]

(i) Create table: It's a DDL statement of SQL and is used to create a table in the database. It creates an empty structure of the table.

Syntax:

Create table < table name> (column1 datatype[(size]), column2 datatype[(size]),
column3 datatype[(size)],……);

Example:

Create table employee (empno number(5), empname varchar2(20),Salary number(8,2));

This statement will create a table 'employee'with the structure containing 3 fields as empno with size for accepting integer data as 5, empname with maximum 20 characters and salary containing float values upto 2 decimal places.

(ii) Insert : It's a DML statement of SQL and is used to insert data into columns of the table.

Syntax :

Insert into <tablename> [column1,column2, column3,….] values (value1,value2, value3,…);

Column names in the syntax are compulsory in case where there is a variation in number of column names and/or sequence of values.

Example :

Consider employee schema as {empno, empname, salary} then

1) Insert into employee values(101, 'abc',12000.50);
2) Insert into employee (empno,empname) values(102,'xyz');

5. Attempt any four: [Marks 16]

a) Explain any four aggregate functions with suitable examples.

[Any 4 functions, Each function 1 Mark]

Following are four aggregate functions:

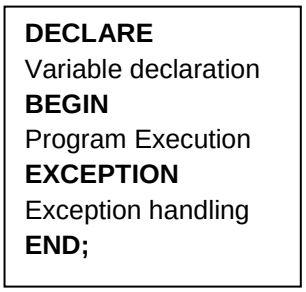
- min
- max
- avg
- sum

- count.

min	**It returns the smallest value in a given column** Example: SELECT min(ssc_per) from stud_info;
max	**It returns the largest value in a given column** Example: SELECT max(ssc_per) from stud_info;
sum	**It returns the sum of the numeric values in a given column** Example: SELECT sum(fees) from branch_details;
avg	**It returns the average value of a given column** Example: SELECT avg(ssc_per) from stud_info;
count	**It returns the total number of values in a given column** Example: SELECT count(stud_id) from stud_info;
count(*)	**It returns the number of rows in a table** Example: SELECT count(*) from stud_info;

b) Draw and explain in brief PL/SQL block structure.

[Diagram 1 Mark, Explanation 3 Marks]



Each PL/SQL program consists of SQL and PL/SQL statements which from a PL/SQL block.

A PL/SQL Block consists of four sections:

The Declaration section : Declaration of memory variables used later in begin section.

The Begin..End section: SQL executable statements for manipulating table data should be in BEGIN....END block.

The Exception section : SQL and/or PL-SQL code to handle errors that may crop up during execution of the above code block.

c) Compare different data models.

[For each point 1 Mark, 4 points should be there]

{Note: comparison of any two models out of three can be considered.}

Relational model	Network Model	Hierarchical Model
Supports one to many and many to many relationships.	Supports many to many relationships.	Supports one to many relationships.
This is table based model i.e it is collection of rows and columns.	It is based on records and links.	It is based on tree like structure with one root.
It is more popular.	Less popular	Less popular
There are many applications of the relational model which are unlimited	Network model is upgraded version of the hierarchical model so used in the networks.	The main application of hierarchical data model is in the mainframe database system.
Data storage is in the form of tables	Data storage is in the form of arbitrary graphs.	Data storage is in the form of data tree i.e. parent child relationship
Does not use pointers or links to relate data, instead uses values to relate data	Uses links to relate data	Uses pointers to relate data

d) What are the functions of DBA ?

[Any 4 functions, 1Mark each]

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.

1. Taking backup of database periodically
2. Ensuring enough disk space is available all the time.
3. Monitoring jobs running on the database.
4. Ensure that performance is not degraded by some expensive task submitted by some users.

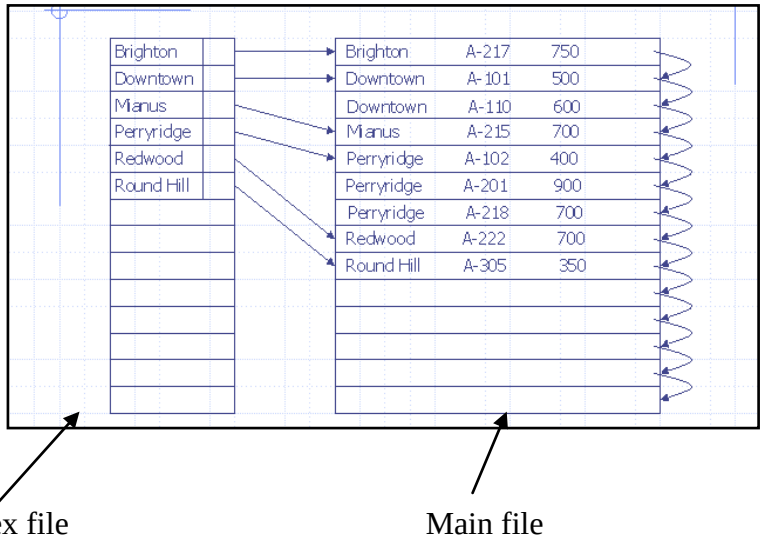
e) How to access records using sequential index access method?

[Explanation 4 Marks]

- Sequential file that is indexed is called as index sequential file.
- The records are stored sequentially by primary key values and there is an index built over the primary key field.
- An index is a set of <key, pointer>pairs where the pointer is the position in the data file of the record with the given key.

- The key search proceeds as follows

  The search key is compared with the index(value) to find the highest index key preceding the search one and a linear search is performed from the record.
- The index key points onwards until the search key is match or until the record pointed by the next index entry is reached.
- In spite of the double file access(index+data) needed by this kind of search the decrease in access time with respect to a sequential file is significant.
- The index allows for random access to records, while the sequential storage of the records of the file provides easy access to the sequential records.



f) Describe the term decomposition and loss less join decomposition.

[Explanation decomposition – 2 Marks, Explanation loss less join decomposition – 2 Marks]

Decomposition replaces an "unnormalized" relation by a set of normalized relations

For relation R(x,y,z) there can be 2 subsets: R1(x,z) and R2(y,z)

If we union R1 and R2, we get R

R = R1 U R2

If **R** is a relation scheme then **{R_1, R_2, ..., R_n}** is a decomposition if

$R = R_1 \cup R_2 \cup ... \cup R_n$

Lossless Decomposition:

A decomposition can be defined as a relation schema R which is replaced by a set of relation schemas {R1,R2,R3,....Rn} such that R1 U R2 ....U Rn=R by decomposing a given relation into the collection of

schemas some of the anomalies in the original relation can be eliminated. This decomposition should be lossless such that no information is lost in the process.

Example of lossless decomposition:

Consider emp schema as { empno, ename, desg}. If the schema is decomposed as

emp1 ={empno, desg } and emp2={empno, ename} then it's a lossless decomposition as considering each sub relation, there exists no loss of data.

6. Attempt any two: [Marks 16]

a) What is transaction? List its properties and explain in brief any two properties.

[Transaction definition 2 Marks, Listing of properties 2 Marks and Explanation of any 2 properties 2 Marks]

A transaction is a logical unit of work of database processing that includes one or more database access operations.

OR

A single logical unit of a work which is the collection of several operations is called as transaction.

Transaction should possess several properties. They are often called as ACID properties & they should be enforced by the concurrency control & recovery methods of the DBMS.The following are the ACID properties:

1) Atomicity: A transactions is an atomic unit of processing; it is either performed entirely or not performed at all

2)Consistency :The consistency property of a transaction implies that if the database was in consistent state before the start of a transaction then on termination of the transaction the database will also be in consistent state.

3) Isolation: Even though multiple transactions may execute concurrently each transaction is unaware of the other transaction executing concurrently in the system. This property of transaction indicates that the action performed by a transaction will be hidden from the other transactions until the transaction terminates.

4) Durability: It states that changes applied to the database by committed transaction must persist in the database. The durability property of a transaction ensures that once the transaction completes successfully the changes it has made to the database remains even if there is system failure.

b) What are types of deadlock protection? Explain any one.

[Explanation of deadlock protection/prevention 2 Marks, Listing Two types 1Mark each, 4Marks for explanation of any one type]

Deadlock Protection/Prevention:

Deadlock prevention protocols ensure that system will never enter into a deadlock state.

One method is each transaction locks all its data items before it begins execution i.e predeclaration.

There are two approaches to deadlock prevention. The approach ensures that no cyclic waits can occur by ordering the request for locks, or requiring all locks to be acquired together. The other approach is closer to deadlock recovery & performs transaction rollback instead of waiting for a lock, whenever the wait could potentially result in a deadlock.

The different deadlock prevention schemes:

1) **The Wait and Die scheme** is a non pre-emptive technique. When transaction Ti requests a data item currently held by Tj,Ti is allowed to wait only if it has timestamp smaller than that of Tj.Otherwise Ti is rolled back which means if the older transaction wants the lock it will wait for the other transaction to complete,if the younger transaction want the lock it will die(roll itself back) It will be rescheduled using the same timestamp.

2) **The Wound –Wait scheme** is pre-emptive technique. It is a counterpart to the wait-die scheme. When transaction Ti requests a data item currently held by Tj,Ti is allowed to wait only if it has a timestamp larger than that of Tj(i.e Ti is younger than Tj).Otherwise Tj is rolled back which means if the older transaction wants the lock it will rolled back the younger transaction. If the younger transaction wants the lock it will wait until the other transaction is completed.

c) Explain general strategies for query processing.

[Any 4 basic strategies 2Marks each]

{Note: If steps of Query Processing is written, considering optimization part give marks accordingly}

The main aim of the query processing is to minimize cost of each query execution. The cost may be in the form of time or space complexity.

Thus query processing to reduce the size of the intermediate & final results as well as processing cost.

The general strategies are as follows:

1) **Perform selection as early as possible:**

Selection reduces the cardinality of the relation & reduces the subsequent processing time.

2**) Combine number of unary operations:** Both the selection & projection can be done on the tuple or relation R simultaneously, requiring a single pass over the tuples,thus providing singular access to them.

For example π_x $(\sigma_y(R))$ i.e Both selection and projection operations can be done simultaneously.

Similarly $\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c1 \wedge c2}(R)$ Where c1 and c2 are certain conditions.

3) **Conversion of Cartesian into join**:

Convert the Cartesian with a certain subsequent selection into a natural join.

e.g. $\sigma_y (R \times S) = R \bowtie S$

4) **Compute common expression only once:**

A common expression that appears more than once in query may be computed once, stored & then reused.

5**) Preprocess the relation**:

Before performing an operation such as a join we can preprocess the relation. The preprocessing includes sorting & index creation on the join attributes.