UNIT - 2

1 Mark

1) Define the terms : Relational Databases, Tables.

A:- Relational Database :- An RDBMS is a type of database Management system (DBMs) that stores data in a row-based table structure which connects related data elements. An RDBMS includes functions that maintain the security, accuracy, integrity and consistency of the data. This is different than the file storage used in a DBMS.

Tables :- In DBMS data is stored in the form of relations i.e., in the tables. A table is a database object which is used to store data in relational database in the form of rows and colums.

2) Explain Views in SQL Language.

A:- Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database. A view can either have all the rows of a table or specific rows based on certain condition.

3) Explain integrity constraints over relations.

A:- * IC : condition that must be true for any instance of the database ; e.g., domain constraints.

* ICs are specified when schema is defined.

* ICs are checked when relations are modified.

* A legal instance of a relation is one that satisfies all specified ICs.

* DBMS should not allow illegal instances.

* If the DBMS checks ICs, stored data in more faithful to real-world meaning.

* It Avoids data entry errors, too!

4) List the primitive Operators in Relational algebra.

A:- Selection Operator (σ): Selection operator is used to selecting tuples from a relation based on some condition.

Syntax: σ (cond) (Relation Name)

Projection Operator (π): Projection operator is used to project particular columns from a relation.

Syntax:- π (column 1, column 2.... column n) (Relation name)

5) What is a key constraint?

A:- * These are called uniqueness constraints since it ensures that every tuple in the relation should be unique.

* A relation can have multiple keys or candidate keys (minimal superkey), out of which we choose one of the keys are as the primary key, we don't have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key.

UNIT – II

3 MARKS

1> Explain about various integrity constraints in relational model ?

Ans: * The DBMS must therefore help prevent the entry of incorrect information.

* An integrating constraints is a Condition specified on A data base schema and restricts the data that can be stored in an instance of data base.

1> Key constraints :

* It don't accepts the duplicate data (or) existing assets.

* Construct the students relation on the constraint that no two have the same student ID.

* This integrating key constraint is a statement that a certain minimal subset of the fields, of a relation is a unique identifier for a tuple.

Eg: Create table student (sid. char(20),
name char (30),
login char(20),
age integer,
gpa real,
unique (name, age),
Constraints: primary key (sid).

2> Foreign key Constraint :

set of fields in one relation that is used to 'refer' to relation.

cid	grade	student
Constraints no	C	53831
Reggae 202	B	53628
Togology 112	A	53650
History 105	B	53666

sid	name	login	age	gPA
50000	Dare	dare@cs	19	8.3
53666	John	John@cs	18	3.4
53688	smith	smith@cs	18	3.2
53650	smith	smith@cs	19	3.8
53831	Madayan	madayan@cs	11	1.8
53628	Gudidee	Gulidee@cs	12	2.0

2) Explain domain relational Calculus ?

Ans: A domain variable is a variable that ranges over the values in the domain of some attribute. (e.g the variable can be assigned an integer if it appears in an attribute whose domain is the set of integers). A DRC query has the form $\{(x_1, x_2 \ldots x_n) / P(\{x_1, x_2 \ldots x_n)\}$, where each x_i is either a domain variable or a constant and $P(x_1, x_2 \ldots x_n)$ denotes a DRC formula whose only free variable are the vari-ables among the x_i, $1 \leq i \leq n$. The result of this query, is the set of all tuples $(x_1, x_2 \ldots x_n)$ for which the formula evaluates to true.

1) Find the names of Sailors who have reserved boat 103.

$(N) \backslash \exists I, T, A\ (I, N, T, A) \in$ Sailors.

$\wedge \exists Ir, Br, D\ (Ir, Br, D) \in$ Reserves $\wedge Ir = I \wedge Br =$.

2) Find the names of Sailors who have reserved a red boat.

$\{(N) \backslash \exists I, T, A\ (I, N, T, A) \in$ Sailors.

$\wedge \exists I, Br, D \in$ reserves $\wedge \exists\ (Br, BN, \text{'red'}) \in$ Boats$)\}$

3) Find the names of sailors who have reserved at least two boats.

$(N) \backslash \exists I, T, A\ (I, N, T, A) \in$ sailors $\wedge \exists Br1, Br2, D1, D_2\ (I, Br_1, D_1)$

$\in$ reserves $\wedge (I, Br_2, D_2) \in$ Reserves $\wedge Br1 = Br2)$

4) Find the names of sailors who have reserved all boats.

$(N) \backslash \exists I, T, A\ (I, N, T, A) \in$ Sailors $\wedge \exists B, BN, C(\neg (B, BN, C) \in$ Boats$) \wedge$

$(\in (Ir, Br, D) \in$ Reserves $(I = Ir \wedge Br = B))))\}$

3) write brief notes on altering tables and views ?

A: The alter table is used to add, remove, or modify columns in an existing table.

* ADD Column: ADD is used to add columns to the existing

table.

Example: ALter table students

ADD Email varchar(225);

* Drop column : Drop column is used to drop columns in a table. Deleting the unwanted coloumns from the table

Ex: Drop column Email.

* Modify column: It is used to modify the existing columns in a table. multiple columns can also be modified at once.

Ex: MODIFY COLUMN : column_name datatype;

⇒ 3) Views : views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. we can create a view by selecting fields from one or more tables present in the database. A view can either have all the rows of a table or specific rows based on the certain condition.

4) write short notes on difference, union, rename and cartesian product operations in relational algebra?

Ans: Difference: Difference (-) is used to retrieve the tuples which are present in R but not in S(R-S).

Union (U): union is used to retrive all the tuples from two relations.

Rename : The rename operator ρ is one of the unary operators in relational algebra and is used to rename relations in a DBMS.

Cartesian product: Cartesian product (x) is used to combine each tuple from the first relation with each tuple from the second relation

5) Explain about Querying relation data?

Ans A relational data base Query that is a Question about the data, the answer consist of new relation Containing the result.

⇒ A query language is a specilised language for writing querys.

⇒ Consider the instance of students relation, we can retrive rose corresponding to students who are younger than 18. with the following SQL Query.

```
SELECT*
FROM Students .S
WHERE S.age <18.
```

⇒ The symbol * means that we retain all fields of selected tuples in the result.

⇒ Think of `S' as a variable that takes on the value of each tuple in students, one tuple after the other.

⇒ The condition S age<18. where class specifies that we want to select only tuples in which the age field has a value has less than 18.

⇒ This query evaluates to the relation shown in given below table.

sid	Name	Login	age	gpa
53831	Madayan	madayan @cs	11	1.8
53628	Gulidee	Gulidee @cs	12	2.0

Students with age less than 18 on instance S_1.

⇒ In addition subset of tuples, query can extract A subset of the fields of each selected tuple we can Compute the names and logins of the students who are younger than 18 with the following query.

Sname	s.login.
Madayan	madayan@cs
Gutidee	Gulidee @cs

Table: names ;login of the student under 18.

UNIT-2 (5 MARKS)

1 Explain about outer join operation in relational algebra.

Ans The join operation is one of the most useful operations in relational algebra and is the most commonly used way to combine information from two or more relations. Although a join can be defined as a cross-product followed by selections and projections, join arise much more frequently in practice than plain cross-products. Joins have received a lot of attention, and there are several variants of the join operations.

Conditional Joins

The most general version of the join operation accepts a join condition c and a pair of relation instances as arguments, and returns a relation instance. The join condition is identical to a selection condition in form. The operation is defined as follows:

$$R \bowtie_c S = \sigma_c(R \times S)$$

Thus, $\bowtie$ is defined to be a cross-Product followed by a selection. Note that condition c can refer to attributes of both R and S.

Equijoin

A common special case of the join operation

$R \bowtie S$ when the join condition con-sists solely of equalities of the form R.name1 = S.name2, that is, equalities between two fields in R and S In this case, obivously, there is some redundancy in retaining both attributes in the result.

Natural Join

A further special case of the join operation $R \bowtie S$ is an equijoin in which equalities are specified on all fields having the same name in R and S. In this case, we can simply omit the join condition; the default is that the join condition is a collection of equalities on all common fields.

2. Explain about domain relational calculus with example.

Ans. Domain relational calculus (DRS) is a calculus that was introduced by Michel Lacroix and Alain Pirotte as a declarative database query language for the relational data model.

In DRS, queries have the form:

$\{ \langle x_1, x_2, ---- x_n \rangle \mid P \langle x_1, x_2 --- x_n \rangle \}$ where each x_i is either a domain variable or constant and $P(\langle x_1, x_2, ---- x_n \rangle)$ denotes a DRC formula. trueThe result of the query is the set of tuples x_1 to x_n that make the DRC formula true.

Example

Table-1: customer

customer name	street	city
Debomit	Kadamtala	Alipurduar
Sayantan	Udaypur	Balurghat
Soumya	Nutanchati	Bankura
Ritu	Juhu	Mumbai

Table 2: Loan

Loan number	Branch Name	Amount
L01	Main	200
L03	Main	150
L10	sub	90
L08	Main	60

Table 3: Borrower

Customer name	Loan number
Ritu	L01
Debomit	L08
soumya	L03

Query 1 : Find the loan number, branch, amount of loans of greater than or equal to 100 amount.

$\{ \langle l, b, a \rangle \mid \langle l, b, a \rangle \in loan \wedge (a \geq 100)$

Resulting Relation :

Loan number	Branch name	Amount
L01	Main	200
L03	Main	150

Query2: Find the loan number for each loan of an amount greater or equal to 150.

$\{<l> \mid \exists b, a\ (<l, b, a> \in loan \wedge (a \geq 150)$

Resulting relation :

Loan number
L01
L03

3. Compare between super key, Candidate key, Primary key for a relation with examples.

Ans Super key

Super key is an attribute that is used to uniquely identifies all attributes in a relation. All super keys can't be candidate keys but the reverse is true. In relation, a number of super keys is more than a number of candidate keys.

Example :

We have a given relation R(A, B, C, D, E, F) and we shall check for super keys by following dependencies :

Functional dependencies super key

AB -> CDEF

YES

CD -> ABEF

YES

CB -> DF

NO

D -> BC

NO

By using key AB we can identify the rest of the attributes (CDEF) of the table. similary, key CD. But, by using key CB we can only identify D and F, not A and E. similarly key D.

Candidate key

A candidate key is a set of attributes that uniquely identify the tuples in relation to or table. As we know the primary key is a minimal super key, so there is one and only one Primary key in any relationship but there is more than one candidate key that ~~&~~ can take place. The candidate key's attributes can contain a NULL value which opposes to the primary key.

Example

student {ID, First_Name, Last-name, Age, sex,
phone-no}

Here we can see the two Candidate keys ID and {First-name, last-name, DOB, Phone-no}. so here, there are present more than one candidate keys, which can uniquely identify a tuple in a relation.

Primary key

The primary key should contain the Unique Values, but can not contain NULL Values. A table can have only one primary key. It is an attribute or a set of attributes that help to uniquely identify the tuples in the relational table.

Example :-

STUDENT-DETAILS

Roll-NO	Name	Marks
101	x	34
102	y	46
103	z	94

Primary key

4. Explain about specifying foreign key constrants in SQL with an Example.

Ans Foreign key Constrants

Sometimes the information stored in a relation is linked to the information stored in another relation. If one of the relations is modified, the other must be stored in stored, checked, and perhaps modified, to keep the data consistent. An Integrating constrants involving both relations must be specified if a DBMS is to make such checks. the most common Integrating constrants involving two relations is a foreign key constrant.

Suppose that, in addition to students, we have a second relation:

Enrolled (studid : string, cid : string, grade: string)

In ensure that only bona fide students can enroll in courses, any value that appears in the studid field of an instance of the Enrolled relation should also appear in the sid field of some tuple in the students relation. the studid field of Enrolled is called a foreign key and refers to students. The foreign key in the referencing relation (Enrolled, in our example) must match that primary key of the referenced relation (students), that is, it must have the same number of columns and compatible data types, although the column names can be different.

cid	grade	studid
carnatic101	C	53831
Reggae203	B	53832
Topology112	A	53650
History105	B	53666

Enrolled (Referencing relation)

Sid	name	login	age	gpa
5000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@math	18	3.8
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Students (Referenced relation).

Fig name: Referential integrity

5. Explain the fundamental operations in relational algebra with enamples.

Ans Relational algebra:

Relational algebra is one of the two formal query languages associated with the relational model.

Set operations:

The following standard operations on set are also available in relational algebra.

1. Union (u)
2. Intersection (n)
3. Set difference (—)
4. Cross product (x)

UNION (∪) R∪S

→ Returns a relation instance containing all tuples that occur in either relation instance R (or) Relation instance S (or) Both.

Example :

S1

sid	sname	rating	age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

sid	sname	rating	age
22	Dustin	7	45.0
31	Luber	8	35.5
28	Luppy	9	35.0
44	guppy	5	35.0
58	Rusty	10	35.0

INTERSECTION.

R∩S returns a relation instance containing all tuples that occur in both R and S. the relation R and S must be union compatible, and the schema of the result is defined to be identical to the schema of R.

Example ÷

sid	sname	rating	age
31	Luber	8	35.5
58	Rusty	10	35.0

Set difference

R-S returns a relation instance containing all tuples that occur. in R but not in S. The Relations R and S must be union compatible. and the schemas of the result is defined to be identical to the schema of R.

Example ÷

$S_1 - S_2$

sid	sname	rating	age
22	Dustin	7	45.0

Cross product

R×S returns a relation instance whose schema contains all the fields of R followed by all the fields of S.

sid	sname	rating	age	sid	bid	day
22	Dustin	7	45.0	22	101	10/10/96
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	22	101	10/10/96
31	Lubber	8	55.5	58	103	11/12/96
58	Rusty	10	35.0	22	101	10/10/96
58	Rusty	10	35.0	58	103	11/12/96

UNIT-2 (10 Marks)

1) What is relation? Differentiate between a relation schema and relation Instance? What are domain Constraints.

A: Any association between two entity types is called a relation.

Difference between relation schema & relation Instance.

Relation Schema	Relation Instance
1) Schema refers to the overall description of any given data base	1) Instance basically refers to a Collection of data and information that the database stores at any particular moment.
2) The schema remains the same for the entire database as a whole	2) One can change the Instances of data and Information in a data-base using. updation, deletion, and addition
3) It does not change very frequently	3) It changes very frequently
4) We use schema for defining the basic structure	4) We use instance for referring to a set of

of any given database. It defines how the available needs to get stored.	Information at any given Instance/time

Domain Constraints

Domain Constraints in DBMS are the set of rules which defines what kind of attributes can be stored in an entity (a table that stores data). Domain Constraints helps us to enter the data into the table according to the particular data type.

Domain Constraints specify two things - Data Type and Constraints such as NOT NULL, PRIMARY KEY, FOREIGN KEY, CHECK, etc. In other words, these constraints define the set of rules (domain) for an attribute of an entity. That's why these are called domain Constraints. This Constraint also ensures that the value taken by an attribute is an atomic value. This means that it cannot be divided from its domain.

Q) What are Integrity Constrains ? Define the terms Primary key Constraints and foreign key Constrains. How are these expressed in SQL?

A:- Integrity constrains can be defined as a set of rules that are used to maintain the information's quality. This ensures that the data integration is not affected at all by data insertion, updation on other processes. Hence, Integrity constraints are like insurance to guard the database if there is any accidental damage.

Integrity Constraints can be of four types:-

1) Domain Constraint:- Domain Contraints Can be defined as a set of values that are valid for an attribute. The domain's data type includes string, string, character, integer, time etc. The value must be in the corresponding domain of the attribute.

2) Entity Integrity Constraint:- This Constraint states that in DBMS we can not make the Primary key with the value NULL. Now, this is because if the primary key is NULL, then we won't able to determine or identify the tuple in the relation.

But in a relation, there Can be NULL values but they must be not the Primary Key

3) Referential Integrity Constraint:- This constraint is defined between two tables. Let us Consider tables A and Bo, so in this constraint, if a foreign key is referred to as a primary key of another table then the Contents of the foreign key of table A must be null or available to tp table B.

4) Key Constraint:- In DBMS, a key is used to uniquely identify an entity in an entity set. An entity set can have multiple keys but out only one can be primary key. This primary key should not be null and must be unique.

Although it Can Contain a null and a non-null unique value.

Primary Key Constraint & its Expression in SQL

A Primary key is used to ensure that data in the Specific column is unique. A column Cannot have NULL values. It Is either an existing table column or a column that is Specifically generated by the database according to a defined Sequence.

Example: STUD_No, as well as STUD_PHONE both, are candidate Keys for relation STUDENT but STUD_No can be choosen as the Primary key.

Foreign key Constraint & its Expression in SQL

A foreigh key is a Column or group of columns in a relational database table that provides a link between data in two tables. It is a column that references a column of another table.

Example:- STUD_No in STUDENT_COURSE is a foreign key to STUD_NO in STUDENT relation.

3) Explain Tuple relational calculus

Tuple Relational Calculus (TRC) is a non-Procedural query language used in relational database management systems (RDBMS) to retrieve data from tables. TRC is based on the Concept of tuples, which are ordered sets of attribute values that represent a single row or record in a database table.

TRC is a declarative language, meaning that it specifies what data is required from the database, rather than how to retrieve it. TRC queries are expressed as logical formulas that describe the desired tuples.

Syntax: The basic syntax of TRC is as follows:
$\{t \mid P(t)\}$
Where t is a tuple variable and p(t) is a logical formula that describes the Conditions that the tuples in the result must satisfy. The curly braces { } are used to indicate that the expression is a set of tuples.

For Example:- Let's say we have a table called "Employees" with the following attributes:

Employee ID
Name
Salary
Department ID

To retrieve the names of all employees who earn more than $50,000 per year, we can use the following TRC query:

$\{t \mid \text{Employees}(t) \wedge t.\text{Salary} > 50000\}$

In this query, the "Employees(t)" expression specifies that the tuple variable t represents a row in the "Employees" table. The "∧" Symbol is the logical AND Operator, which is used to Combine the Condition "t.salary > 50000" with the table selection.

The result of this query will be a set of tuples, where each tuple Contains the Name attribute of an Employee who earns more than $50,000 per year.

TRC can also be used to perform more Complex queries, such as joins and nested queries, by using additional logical operators and expressions.

While TRC is a Powerful query language, it Can be more difficult to write and understand than other SQL-based query languages, such as structured Query language (SQL).

Tuple Relational Calculus is a non-procedural query language, Unlike relational algebra. Tuple Calculus provides only the description of the query but it does not provide the methods to solve it. Thus, it explains what to do but not how to do it.