

UNIT III

Relational Model: Introduction, CODD Rules, relational data model, concept of key, relational integrity, relational algebra, relational algebra operations, advantages of relational algebra, limitations of relational algebra, Functional dependencies and normal forms upto 3rd normal form.

Q) What is a Relational Data Model

1. The relational model uses a collection of tables to represent both data and relationships.
2. Tables are logical structures maintained by the database manager.
3. The relational model is a combination of three components,
 - a. Structural
 - b. Integrity and
 - c. Manipulative parts.

a. Structural Part

1. The structural part defines the database as a collection of relations.

b. Integrity Part

1. The database integrity is maintained in the relational model using primary and foreign keys.

c. Manipulative Part

1. The relational algebra and relational calculus are the tools
2. It used to manipulate data in the database.
3. Thus relational model has a strong mathematical background.

Table: (Characteristics of a relational table)

A table is perceived as a two-dimensional structure composed of rows and columns.

1. A table is also called relation.
2. Each row in the table is called tuple.
3. Each column in the table is called attribute.
4. Each column represents an attribute, and each column has a distinct name
5. Each rows and columns can be exactly the same
6. The column will have data value.
7. All values in a column must conform to the same data format
8. The order of the rows and columns is unimportant to the DBMS.
9. In relational model attributes can be in any order.
10. Relations must have a key.
11. Keys can be a set of attributes.
12. Each column has a specific range of values known as the attribute domain
13. Domain is the set of valid values for an attribute.
14. Degree of the relation is the number of attributes (columns) in the relation.
15. Cardinality of the relation is the number of tuples (rows) in the relation

Q) Explain Dr. E.F.CODD Rules

[For any RDBMS to be accepted as full fledged RDBMS it has to follow the 12 CODDrules . The rules are] There are 12 CODD rules.

1. **The information Rule:** it is used to be represented all information as a data values in the rows and columns of tables. This is the basis of the relational model.
2. **The guaranteed access Rule:** Every data value in a relational database should be logically accessible by specifying a combination of the table name, the primary key value on column name.
3. **Systematic Treatment of NULL values:** DBMS must support NULL values to represent missing or inapplicable information. They must be distinct from zero or spaces.
4. **Active Online Catalog Based on the Relational model:** The system list is a collection of tables that the DBMS maintains for its own use. These tables hold the description of the structure of the database.
5. **The Comprehensive Data Sub-language Rule:** This Rule states that the system must support at least one language that performs data definition, View Definition, data manipulation operations, security and integrity constraints and transaction management operations.
6. **The View Updation Rule:** It used to All views that are theoretically updateable must be updateable by the system.
7. **High Level insert, update, and Delete:** it used to rows should be treated as sets in insert, delete and update operations.
8. **Physical data independence:** Application Program must remain un impaired (undamaged) when any changes are made in storage representation or access method.
9. **Logical Data Independence :**Users and user Programs should be independent of the logical structure of the database.
10. **Integrity independence:** Integrity constraints must be storable in System catalog.
11. **Distribution independence:** Database must allow manipulation of Distributed data located on other computer systems.
12. **Non – Subversion Rule:** This rule states that different levels of the language can not subvert or bypass the integrity rules and constraints.

Q) Types of SQL Keys

There are different types keys relation data model. They are

1. Composite key
2. Candidate key
3. Super key
4. Alternate key
5. Secondary key
6. Primary key
7. Foreign key

1. Composite key:

- a. It is a combination of two or more columns in a table
- b. That can be used to uniquely identify each row in the table.
- c. Such a multi attribute key is known as composite key.

Ex:

```
CREATE TABLE BABA (Sno integer, deptno integer, address varchar(50),
PRIMARY KEY (sno, deptno));
```

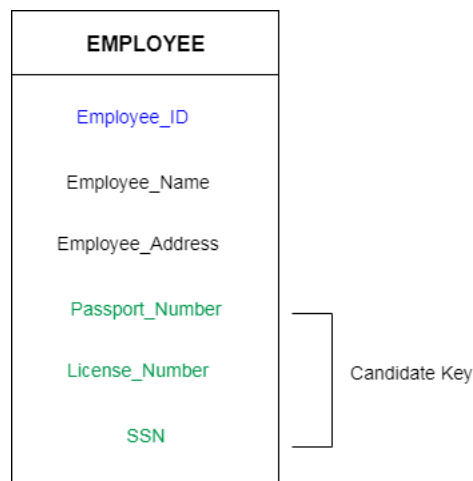
Explanation:

2. Candidate key:

- a. A **Candidate Key** can be any column or a combination of columns
- b. That can qualify as unique **key** in database.
- c. There can be multiple **Candidate Keys** in one table.
- d. Each **Candidate Key** can qualify as Primary **Key**.

[A Primary **Key** is a column or a combination of columns that uniquely identify a record
A minimal super key. A super key that does not contain a subset of attributes that is itself a super key.]

For example: In the EMPLOYEE table, id is best suited for the primary key. Rest of the attributes like SSN, Passport_Number, and License_Number, etc. are considered as a candidate key.



3. Super key:

- Super key** is a combination of columns
- That uniquely identifies any row within a RDBMS table.
- It is reduced to the minimum number of columns required to uniquely identify each row.
- An attribute or combination of attributes that uniquely identifies each row in a table is called super key.

For example: In the above EMPLOYEE table, for(EMPLOYEE_ID, EMPLOYEE_NAME) the name of two employees can be the same, but their EMPLOYEE_ID can't be the same. Hence, this combination can also be a key. The super key would be EMPLOYEE-ID, (EMPLOYEE_ID, EMPLOYEE-NAME),

4. Alternate key:

- A Alternate key is a key that can be work as a primary key.
- Basically it is a candidate key that currently is not primary key.

Or

Among of candidate keys if any single key or combination of keys made as primary key then rest candidate key called as alternate key.

Exp:

- Suppose in employee table EmpID is primary key
- Then Emailid and SSN are called as alternate key
- It mean later on these key can be act as primary key without affecting existing data in table.

Secondary key:

1. An attribute (or combination of attributes) used strictly for data retrieval purpose is called secondary key.

Exp :

Create table tnam(ano number, bno number, bname char(12), primary key(ano,bno));

In this bno is secondary key.

Primary key:

1. An attribute that uniquely identifies each row in a relation is called primarykey.
2. The combination of unique and not null
3. It cannot support duplicate value and also empty value in attribute

Ex: CREATE TABLE BABA

(Sno integer, deptno integer, address varchar(50), PRIMARY KEY (sno));

Or

CREATE TABLE BABA

(Sno integer primary key, sna varchar2(20));

Foreign key:

1. It is used to create relation between two tables using with primary key.
2. A FOREIGN KEY is a key used to link two tables together
3. A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.
4. The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

```
CREATE TABLE BABA1 (
  dno int PRIMARY KEY,
  dname varchar2(20) NOT NULL,
  Sno integer FOREIGN KEY REFERENCES BABA(Sno)
);
```

Q) Integrity rules or relational integrity:

1. Relational database integrity rules are very important to good database design.
2. Many (but by no means all) RDBMSs enforce integrity rules automatically.
3. The set of rules which are used to maintain the accuracy of the data and to integrate (add) the data in database is called as integrity constraints.
 1. Domain Integrity.
 2. Entity Integrity.
 3. Referential Integrity.

1. Domain Integrity:

1. A domain is a set of values which is assigned to an attribute.
2. The components of the domain are domain name, meaning, data type, size and range of the values.

Ex: student relation the domains are regdno, name.

2. Entity Integrity :

1. All primary key entries are unique, and no part of a primary key may be null.
2. Each row will have a unique identity, and foreign key values can properly reference primary key values.
3. A primary key is an attribute in a relation which is used to identify each row in a relation.
4. No. of components of a primary key value may be NULL.
5. This is called as entity integrity.
6. A NULL is an undefined value. One NULL is not equal to another NULL.

3. Referential Integrity :

1. A referential integrity is used to implement the foreign key.
2. A foreign key is a non key attribute in the relation R and it is a primary key in relation R2.
3. This key is called as foreign key. The value of the foreign key is either NULL or primary key.
4. Another integrity rules that can be enforced in the relational model are the **NOT NULL** and **UNIQUE** constraints.

NOT NULL: it does not support empty value in attribute. The NOT NULL constraint can be placed a value for that column.

UNIQUE: The UNIQUE constraint is used to not support duplicate values exists column.

Q) What is a Relational Algebra

1. The relational algebra is a notional language with operations
2. That work on one or more relations to define another relation without changing the original relation.
3. The both the operands and the results are relations;
4. The output from one operation can become the input to another operation.
5. This allows expressions to be nested in the relational algebra. This property is called closure.
6. Relational algebra is an abstract language,
7. It means that the queries formulated in relational algebra are not intended to be executed on a computer.
8. Relational algebra consists of group of relational operators that can be used to manipulate relations to obtain a desired result.
9. Knowledge about relational algebra allows us to understand query execution and optimization in relational database management system.

Q) Explain Relational Algebra Operations (Relational Set operators)

Operations in relational algebra can be classified into set operation and database operations.

1. Unary and Binary Operations

1. Unary operation involves one operand,
 - a. selection
 - b. Projection

2. Binary operation involves two operands.

- a. Union operation
- b. Difference operation
- c. Cartesian product operation
- d. Intersection operation
- e. Division operation
- f. Join operations operation

2. Relational algebra operations

3. Set Operations Database operations

Three main database operations are SELECTION, PROJECTION, and JOIN.

a. Selection Operation

1. The selection operation works on a single relation R
2. And it defines a relation that contains only those tuples of R
3. That satisfies the specified condition (Predicate).
4. Selection operation can be considered as row wise filtering.
5. This is pictorially represented also known as RESTRICT
6. It creates values for all rows found in a table that satisfy a given condition.
7. Select can be used to list all of the row values, or it can create only those row values
8. That matches a specified criterion.
9. Select create a horizontal subset of a table.

Syntax : σ Predicate (R).

ex: σ subject = "database"(Books)

P_CODE	P_DESCRIPT	PRICE
123456	Flash light	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
123459	9v battery	1.92
123455	100W bulb	1.47
123454	Powerdrill	34.99

SELECT only PRICE less than Rs. 2.00 yields

P_CODE	P_DESCRIPT	PRICE
123459	9v battery	1.92
123455	100W bulb	1.47

b. Projection Operation

1. The projection operation works on a single relation R
2. And defines a relation that contains a vertical subject of R,
3. And extracting the values of specified attributes and elimination duplicates.
4. The projection operation can be considered as column wise filtering.

It's creates all values for selected attributes. In other words, project creates a vertical subset of a table.


Syntax : $a_1, a_2, \dots, a_n (R)$.

[Where a_1, a_2, \dots, a_n are attributes and R stands for relation To illustrate projection operation consider the relation STAFF, with the attributes Staff number, Name, Gender, Date of birth, and Salary.]

SALES:

P_CODE	P_DESCRIPT	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
123459	9v battery	1.92
123455	100w bulb	1.47
123454	Power drill	34.99


PROJECT PRICE yields



Select price from sales;

PRICE
5.26
25.15
10.99
1.92
1.47
34.99

PROJECT P_DESCRIPT and PRICE yields



Select components, price, from sales;

P_DESCRIPT	PRICE
Flashlight	5.26
Lamp	25.15
Box Fan	10.99
9v battery	1.92
100w bulb	1.47
Power drill	34.99

a. Union Operation

1. Union combines all rows from two tables, excluding (without) duplicate rows.
2. The tables must have the same attribute characteristics to be used in the UNION.

Or

3. The union of two relations R and S defines a relation
4. That contains all the tuples of R or S or both R and S,
5. Duplicate tuples being eliminated.

Relational Algebra Expression

The union of two relations R and S are denoted by $R \cup S$. $R \cup S$ is pictorially represented

P_CODE	P_DESCRIPTION	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
123459	9v battery	1.92
123455	100w bulb	1.47
123454	Powerdrill	34.99

UNION

P_CODE	P_DESCRIPTION	PRICE
345678	Microwave	160.00
345679	Dishwasher	500.00

OUTPUT

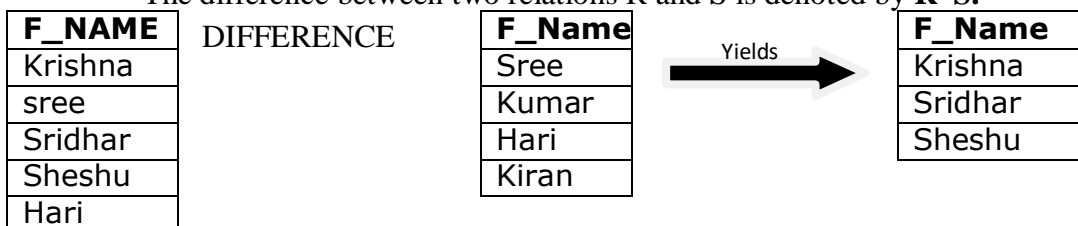
P_CODE	P_DESCRIPTION	PRICE
123456	Flashlight	5.26
123457	Lamp	25.15
123458	Box Fan	10.99
123459	9v battery	1.92
123455	100w bulb	1.47
123454	Powerdrill	34.99
345678	Microwave	160
345679	Dishwasher	500

b. Difference Operation

1. The set difference operation defines a relation consisting of the tuples
2. That is in relation R but not in S.
3. It create all rows in one table that are not found in the other table
4. That is, it subtracts one table from the other.
5. The effect of a DIFFERENCE is shown in follow in example.
6. That subtracting the first table from the second table is not the same as subtracting the second table from the first table.

Relational Algebra Expression

The difference between two relations R and S is denoted by $R - S$.



c. Cartesian Product Operation

1. The Cartesian product operation defines a relation
2. That is the concatenation of every tuples of relation R with every tuples of relation S.
3. The result of Cartesian product contains all attributes from both relations R and S.
4. It's creates all possible pairs of rows from two tables
5. It is also known as the Cartesian product.
6. For example one table has six rows and the other table has three rows,
7. The PRODUCT creates a list composed of $6 \times 3 = 18$ rows.

Relational Algebra Symbol for Cartesian product:

The Cartesian product between the two relations R and S is denoted by **R × S**.

Example

If there are 5 tuples in relation “R” and 2 tuples in relation “S” then the number of tuples in R × S is 5 * 2 = 10.

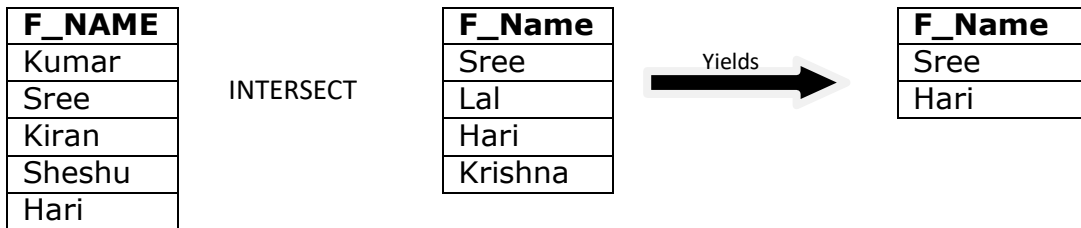
R	S
A	1
B	2
	3

R	S
A	1
A	2
A	3
B	1
B	2
B	3

Note:
No. of tuples in R X S = 2 x 3 = 6

d. Intersection Operation

1. The set of all tuples that are in both R and S.
2. Intersect create only the rows that appear in both tables.
3. As we true in the case of UNION table must be union-compatible to produce(create) valid results.
4. For example, you cannot use INTERSECT if one of the attributes is numeric and one is character – based.



Division operation

The division operator is used for queries which involve the ‘all’.

R1 ÷ R2 = tuples of R1 associated with all tuples of R2.

Example

Retrieve the name of the subject that is taught in all courses.

Name	Course
System	Btech
Database	Mtech
Database	Btech
Algebra	Btech

÷

Course
Btech
Mtech

=

Name
database

The resulting operation must have all combinations of tuples of relation S that are present in the first relation or R.

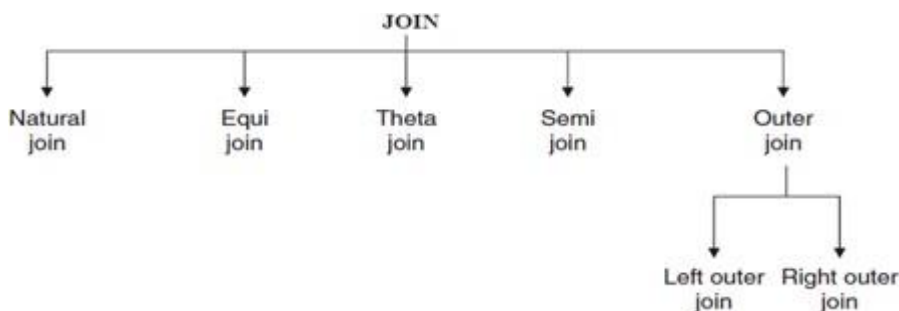
e. Join Operations

1. Join operation combines two relations to create a new relation.
2. The tables should be joined based on a common column.
3. The common column should be compatible in terms of domain.
4. It allows information to be combined from two or more tables.
5. Join is the real power behind the relation database, allowing the use of independent table linked by common attributes (Characteristics).

There are different types of join operations

Types of Join Operation

There are different types of join operations



Theta join

If we join R1 and R2 other than the equal to condition then it is called theta join/ non-equi join.

Example

Consider R1 table

RegNo	Branch	Section
1	CSE	A

RegNo	Branch	Section
2	ECE	B
3	CIVIL	A
4	IT	B
5	IT	A

Table R2

Name	RegNo
Bhanu	2
Priya	4

$R1 \bowtie R2$ with condition $R1.regno > R2.regno$

RegNo	Branch	Section	Name	Regno
3	CIVIL	A	Bhanu	2
4	IT	B	Bhanu	2
5	IT	A	Bhanu	2
5	IT	B	Priya	4

In the join operation, we select those rows from the cartesian product where $R1.regno > R2.regno$.

Join operation = select operation + cartesian product operation

Natural join

If we join R1 and R2 on equal condition then it is called natural join or equi join. Generally, join is referred to as natural join.

Natural join of R1 and R2 is –

{ we select those tuples from cartesian product where $R1.regno=R2.regno$ }

R1 \bowtie R2

Regno	Branch	Section	Name
2	-	-	Bhanu
4	-	-	priya

Outer join

It is an extension of natural join to deal with missing values of relation.

Consider R1 and R2 shown below –

Table R1

RegNo	Branch	Section
1	CSE	A
2	ECE	B
3	CIVIL	A
4	IT	B
5	IT	A

Table R2

Name	Regno
Bhanu	2
Priya	4
Hari	7

Outer join is of three types. These are explained below –

Left outer join

It is denoted by $R1 \bowtie R2$.

RegNo	Branch	Section	Name	Regno
2	-	-	Bhanu	2

RegNo	Branch	Section	Name	Regno
4	-	-	Priya	4
1	-	-	NULL	NULL
3	-	-	NULL	NULL
5	-	-	NULL	NULL

Here all the tuples of R1(left table) appear in output.

The mismatching values of R2 are filled with NULL.

Left outer join = natural join + mismatch / extra tuple of R1

Right outer join

It is denoted by $R1 \bowtie R2$

Here all the tuples of R2(right table) appear in output. The mismatching values of R1 are filled with NULL.

RegNo	Branch	Section	Name	Regno
2	-	-	Bhanu	2
4	-	-	Priya	4
NULL	NULL	NULL	Hari	7

Right outer join = natural join+ mismatch/extra tuple of R2.

Full outer join

It is denoted by $R1 \bowtie R2$.

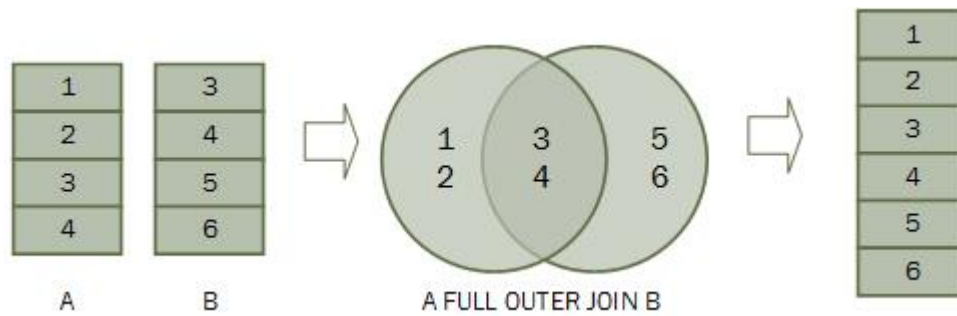
Full outer join=left outer join U right outer join.

RegNo	Branch	Section	Name	Regno
2	-	-	Bhanu	2
4	-	-	Priya	4
1	-	-	NULL	NULL
3	-	-	NULL	NULL

RegNo	Branch	Section	Name	Regno
5	-	-	NULL	NULL
NULL	NULL	NULL	Hari	7

Example

Given below is the picture of full outer join –



Advantages and Limitations

Advantages

- Relational algebra is based on the set theory which is a mathematical concept due to which it has a scope of development.
- Like mathematics there can be many expressions for the same operation, in a similar way if there are two relational algebraic expressions for the same operation then the query optimizer will switch to the most efficient query.
- It is a high-level query language.

Limitations

- Relational algebra cannot perform arithmetic operations.
- It is unable to do aggregation operations even it cannot compute transitive closure.
- It cannot modify the data present in the database.