

Assignment-II

B. Koushik Reddy
2211CS010052
Group-04

1, Explain about Integrity constraints over relations?

An Integrity constraint (IC) is a condition that is specified on a database schema and restricts the data can be stored in an instance of the database. Various restrictions on data can be specified on a relational database schema in the form of 'constraints'. A DBMS enforces integrity constraints, in that it permits only legal instances to be stored in the database. Integrity constraints are specified and enforced at different time as below.

1. When the DBA or end user defines a database schema, he or she specifies the IC's that must hold on any instance of this database.

2. When a database application is run, the DBMS checks for violations and disallows changes to the data that violate the specified ICs.

2211CS010052

The constraints can be classified into 4 types as below.

1. Domain Constraints.

Domain constraints are the most elementary form of integrity constraints. They are tested easily by the system whenever a new data item is entered into the database. Domain constraints specify the set of possible values that may be associated with an attribute. Such constraints may also prohibit the use of null values for particular attributes. The data types associated with domains typically include standard numeric data types for integers. A relation schema specifies the domain of each field or column in the relation instance. These domain constraints in the schema specify an important condition that each instance of the relation to satisfy: The values that appear in a column must be drawn from the domain associated with that column.

2. Key Constraints.

A key constraints is a statement that a certain

minimal subset of the fields of a relation is a unique identifier for a tuple.

Example: The 'students' relation and the constraint that no 2 students have the same student id (sid).

Different keys:

- Candidate key
- Super key
- Primary key.

3. Entity Integrity Constraints.

This states that no primary key value can be null. The primary key value is used to identify individual tuples in a relation. Having null values for the primary key, implies that we cannot identify some tuples. NOTE: Key Constraints, Entity integrity Constraints are specified on individual relations. PRIMARY KEYS comes under this.

4. Referential Integrity Constraints

The Referential Integrity Constraints is specified between 2 relations and is used to maintain the consistency among tuples of the 2 relations.

Informally, the referential integrity constraint states that a tuple in 1 relation that refers to another relation must refer to an existing tuple in that relation. We can diagrammatically display the referential integrity constraints by drawing a directed arc from each foreign key to the relation it references.

2) Explain about views with syntax and list out the commands used in views?

Views

A view is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables 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. Views are a logical virtual table created by "Select query" but the result is not stored anywhere in the disk and every time we need to fire the query when we need data, so always we get updated or latest data from original tables.

- A view is a predefined query on one or more tables.
- Retrieving information from a view is done in the same manner as retrieving from a table.
- With some views you can also perform DML operations (delete, insert, update) on the base tables.
- Views don't store data, they only access rows in the base tables.
- user tables, user sequences, and user indexes are all views.
- View can hide the underlying base tables.

Creating Views

A view can be created from a single table or multiple tables or another view.

Syntax:

Simple

```
CREATE VIEW view_name AS SELECT column1,
column2, ... FROM table_name WHERE condition;
```

Complex

```
CREATE VIEW view_name AS SELECT tablename.
attribute1, tablename.attribute2, tablename2.attribute3
... FROM table1, table2 where condition;
```

Note:

A view always shows up-to-date data!

The database engineer recreates the view, every time a user queries it.

DROP View.

SQL allows us to delete an existing view.

We can delete or drop a view using the DROP Statement.

2211CS010052
Syntax: DROP VIEW view_name;

* Name of the view which we want to delete.

Create or Replace

We can use the CREATE OR REPLACE VIEW statement to add or remove fields from a view.

Syntax:

```
CREATE OR REPLACE VIEW view_name AS SELECT  
table1.attribute1, table1.attribute2, table2.attribute3,  
table2.attribute4 FROM table1, table2 WHERE  
Condition;
```

If we fetch all the data from view now as
SELECT * FROM view_name;

Updating Views

The view should be created from a single table.

If the view is created using multiple tables then we will not be allowed to update the view.

- The view should have all NOT NULL values. The view should not be created using nested queries or complex queries.

- 2211CS010052
- If the view you want to update is based upon another view, the latter should be updatable.

Syntax:

```
update view_name set Attribute = value;
```

Inserting a row in a view:

We can insert a row in a view in a same way as we do in a table.

We can use the INSERT INTO statement of SQL to insert a row in a view.

Syntax:

```
INSERT INTO view_name (column1, column2, ----)  
VALUES (value1, value2, ----);
```

Deleting rows from Views:

- Deleting rows from a view is also as simple as deleting rows from a table.
- We can use the DELETE statement of SQL to delete rows from a view.
- Also deleting a row from a view, first delete the row from the actual table and the change is then reflected in the view.

Syntax:

DELETE FROM view_name WHERE condition;

We can fetch all the data from view as

SELECT * FROM view_name;

3) Explain about Selection, projection and rename (Unary operators) with commands?

Unary Relational Operations are.

- SELECT (Symbol: σ)

- PROJECT (Symbol: π)

- RENAME (Symbol: ρ)

1) Selection:

The SELECT operation is used for selecting a subset of the tuples according to a given selection condition.

*> Sigma (σ) is used to denote it.

*> Select Operator select tuples that satisfy a given predicate.

*> Predicate refers to an expression that determines whether something is true or false.

$\sigma P(x)$.

- σ is the select predicate.

- x stands for relation which is the name of the table.

- P is propositional logic formula which may use connectors like and, or, not.

Example 1:

σ topic = "Database" (Tutorials).

Output: Selects tuples from Tutorials where topic = 'Database'.

2) Projection (π):

*> The projection is used to select columns that satisfy a given predicate.

*> The projection method defines a relation that contains a vertical subset of Relation.

*> This helps to extract the values of specified attributes to eliminate duplicate values.

*> (π) Symbol is used to choose attributes from a relation.

Notation:

$$\pi_{A1, A2, \dots, An}(r)$$

where $A1, A2, \dots, An$ are attributes names of relation r .

Consider the following table.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

Here, the projection of CustomerName and Status

$$\pi_{\text{CustomerName, Status}}(\text{Customers})$$

CustomerName	Status
Google	Active
Amazon	Active
Apple	Inactive
Alibaba	Active

3) Rename :

* In relational algebra, a rename is a unary operation written as $\rho_{a/b}(R)$ where :

- a and b are attribute names
- R is a relation.

* It is used for renaming attributes of a relation.

* The result is identical to R except that the b field in all tuples is renamed to an a field.

Example :

Employ

Name	EmployeeID
Harry	3415
Sally	3345

$$\rho_{\text{EmployeeName/Name}}(\text{Employ})$$

Employee Name	EmployeeID
Harry	3415
Sally	3345

4) Explain about different joins with example?

Types of joins:

1. Natural join (\bowtie)
2. Left, Right, Full Outer join (\ltimes , \rtimes , \Join)

Natural Join (\bowtie)

- Natural join is a binary operator \rightarrow (Cross Product + Condition).
- Will result set of all combination of tuples where they have equal common attribute.

Ex:

Emp-code	Emp-name
101	Stephan
102	Jack
103	Harry

Emp-code	Salary
101	50000
102	30000
103	25000

TT Emp-name, Salary (Employee \bowtie salary)

Emp-name	Salary
Stephan	50000
Jack	30000
Harry	25000

Outer Join

- The outer join operation is an extension of the join operation. It is used to deal with missing information.
- Types of Outer join.
 - Left (\ltimes)
 - Right (\rtimes)
 - Full outer join (\Join)

Outer Join - Left

- Left outer join contains the set of tuples of all combinations in R and S that are equal on their common attribute names.
- In the left outer join, tuples in R have no matching tuples in S.

Ex:

Employee.

Emp_name	Street	City
Ram	Civil line	Mumbai
Shyam	Park street	Kolkata
Ravi	M.G. Street	Delhi
Hari	Nehru nagar	Hyderabad

Factworkers

Emp_name	Branch	Salary
Ram	Infosys	10000
Shyam	Wipro	20000
Kuber	HCL	30000
Hari	TCS	50000

Employee \bowtie Factworkers

Emp_name	Street	City	Branch	Salary
Ram	Civil line	Mumbai	Infosys	10000
Shyam	Park street	Kolkata	Wipro	20000
Hari	Nehru street	Hyderabad	TCS	50000
Ravi	M.G. Street	Delhi	NULL	NULL

Outer Join - Right (\bowtie).

- Right outer join contains the set of tuples of all combinations in R and S that are equal on their common attribute names.
- In right outer join, tuples in S have no matching tuples in R.

Ex:

Employee.

Emp_name	Street	City
Ram	Civil line	Mumbai
Shyam	Park street	Kolkata
Ravi	M.G. Street	Delhi
Hari	Nehru nagar	Hyderabad

Factworkers

Emp_name	Branch	Salary
Ram	Infosys	10000
Shyam	Wipro	20000
Kuber	HCL	30000
Hari	TCS	50000

Employee ~~Factworkers~~

Emp_name	Branch	Salary	Street	City
Ram	Infosys	10000	Civil line	Mumbai
Shyam	Wipro	20000	Park street	Kolkata
Harish	TCS	50000	Nehru street	Hyderabad
Kuber	HCL	30000	NULL	NULL

Outer Join - Full

- Full outer join is like a left or right join that it contains all rows from both tables.
- In full outer join, tuples in R that have no matching tuples in S and tuples in S that have no matching tuple in R in their common attribute name.

Ex:-Employee ~~Factworkers~~

Emp_name	Street	City	Branch	Salary
Ram	Civil line	Mumbai	Infosys	10000
Shyam	Park street	Kolkata	Wipro	20000
Harish	Nehru street	Hyderabad	TCS	50000
Ravi	M.G. Street	Delhi	NULL	NULL
Kuber	NULL	NULL	HCL	30000

Self. Join:

- Find Students enrolled in at least two Courses.

- Select T1.S_Id from Study as T1, Study as T2.

where T1.S_Id = T2.S_Id AND T1.C_Id != T2.C_Id.

Ex:

S_Id	C_Id	Since
S1	C1	2019
S2	C2	2020
S1	C2	2020

5) Explain about different commands in SQL (DML, DDL, DCL, TCL).

SQL commands

- 1) (DDL) Data Definition Language.
- 2) Data Manipulation Language (DML).
- 3) Data Control Language (DCL)
- 4) Transaction Control Language (TCL).

1) Data Definition Language (DDL)

The SQL DDL allows specification of not only a set of relations, but also information about each relation, including:

- The schema for each relation.
- The types of values associated with each attribute.
- The integrity constraints.
- The set of indices to be maintained for each relation.

* Basic data types used are, Char, varchar, int, float.

Main Commands,

- Create
- Alter
- Drop
- Rename
- Truncate.

- Create :

Create a new table, a view of a table, or other Object in the database.

Syntax:

CREATE TABLE table_name (Attribute1 datatype, Attribute2 datatype,);

- To create table constraint.

Primary Key: The primary key attributes are required to be nonnull and unique.

CREATE TABLE Persons (PID int(10) NOT NULL, PRIMARY KEY, LastName varchar(25), FirstName varchar(25));

Foreign Key:

Create table orders (OrderID int NOT NULL.

PRIMARY KEY, OrderNumber int NOT NULL, PersonID
int, FOREIGN KEY (PID) REFERENCES Persons(PID);

-ALTER:

Modifies an existing database object, such as a table.

AddSyntax:

ALTER TABLE table_name ADD column_name datatype;

Modify Syntax:

ALTER TABLE tablename MODIFY columnname
datatype;

Delete Syntax:

ALTER TABLE tablename Drop Column column_name;

-DROP:

* Deletes an entire table, a view of a table or other objects in the database.

Syntax:

DROP TABLE table_name;

-RENAME:

* Rename a table or its attribute.

Syntax:

RENAME TABLE tablename TO new-tablename;

-TRUNCATE

* It is used to remove all records from a table.

Syntax:

TRUNCATE TABLE tablename;

2) Data Manipulation Language (DML):

The SQL commands that deal with the manipulation of data present in database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

-SELECT:

It is used to retrieve data from the database.

Syntax:

SELECT * FROM table_name;

-INSERT:

It is used to insert data into a table.

Syntax:

```
INSERT INTO tablename (column1, column2, ----)
VALUES (value1, value2, ----);
```

-UPDATE:

It is used to update existing data.

Syntax:

```
UPDATE table_name SET column1 = value1, column2 =
value2, ---- WHERE condition;
```

-DELETE:

It is used to delete records.

Syntax:

```
DELETE FROM table_name WHERE condition;
```

3) Data Control Language (DCL):

DCL mainly deals with the rights, permissions and other controls of the database system.

-GRANT:

It gives user's access privileges to database.

Syntax:

```
GRANT privilege_name ON Table_name TO user_name;
```

-REVOKE:

It withdraw user's access privileges given by using the GRANT command.

Syntax:

```
REVOKE privilege_name ON Table_name FROM
user_name;
```

4) Transaction Control Language (TCL):

Transaction Control Language (TCL) commands are used to manage transactions in the database. There are used to manage the changes made to the data in a table by DML statements.

-COMMIT:

The command is used to permanently save any transaction into the database.

Syntax:

```
COMMIT;
```

- ROLLBACK :

The command to rollback changes.

Syntax :

SAVEPOINT command. ROLLBACK TO Savepoint_name;

- SAVEPOINT :

The command is used to temporarily save a transaction so that you can rollback to that point whenever required.

Syntax :

SAVEPOINT Savepoint_name;

Transaction Control Language (TCL) :

Transaction Control Language (TCL) commands are used to manage transactions in the database.

There are used to manage the changes made to the data in a table by DML statements.

- COMMIT :

The command is used to permanently save a transaction into the database.

Syntax :

COMMIT ;