UNIT-II

Rational data models -

(1) A relational database consists of a collection of tables, each of which is assigned a unique name

① Domain - Set of permitted values for each attributes

Attributes - Columns or Headers (name)

Tuples - Rows or Records

(table) Relation - Subset of a cartesian product of a list of domains

Characteristics of relation or table -

(1) NO Duplicate tuples

(2) Tuples are unordered

(3) Attributes are unordered.

(4) Attributes values are atomic.

② Keys - A Key allows us to identify a set of attributes that suffice to distinguish ~~entities~~ tuples from each other.

Super Key - Set of one or more attributes that, taken collectively, allows us to identify uniquely a record in the table or relation.

Candidate Key - Having all property of Super Key but not a super key.

Primary Key - It just a super key that is chosen by the database designer means of identifying records or tuples within a table or relation.

Foreign Key - To establish relationship between one two or more than two tables foreign key is used. Foreign Key is a primary key of other relation include in the given relation.

Composite Key - Consist of two or more attributes that uniquely identify an entity occurance.

Secondary or alternate Key - Candidate Key which are not selected for primary Key

③ Schema of a Relation or table -

→ Denoted by R (A1, A2, ... An)

→ R is the name of the relation or table

→ A1, A2, ... An are the attributes of the relation or table

Relational Database Schema -

→ denoted by $S = \{R1, R2, \ldots, Rn\}$

→ S is the name of the whole database schema.

→ R1, R2, ..., Rn are the names of the individual relation schemas within the database S.

④ Integrity Constraints (Relational) -

It is the mechanism or condition used to prevent invalid data entry into the relation (table). It applies rules on the table. Three types of integrity constraints are -

(1) Domain Integrity Constraints - Set a range or any violation that take place will prevent the user from performing the manipulation. Two types -

(1) Not Null → not allow null values

(2) Check → specify condition that each tuple must satisfy.

(2) Entity Integrity Constraints - Applies on relation or table. Two types -

(1) Unique → prevent duplicate values, allow NULL values.

(2) Primary Key → prevent duplicate values, don't allow NULL values.

(3) Referential Integrity Constraints - Applies conditions in two tables.

Foreign Key → Tuples in the referencing relation R1 have attributes (foreign key attributes) that reference the primary key attribute of the referenced relation R2

⑤ Intension - It is a constant value that gives the name, structure of table and the constraints laid on it.

Extension - It is the number of tuples present in a table at any instance. This is time dependent.

Relational Query Languages –

① SQL (Structured Query language) –

SQL was invented and developed by IBM in early 1970's. SQL is used for storing and retrieving information in oracle.

Advantages –

(1) Non-procedural language

(2) Very simple commands for querying, inserting, deleting and updating data.

add, drop, modify columns

② DDL – CREATE, ALTER, DROP, RENAME, TRUNCATE, COMMENT

DML – SELECT, INSERT, UPDATE, DELETE, MERGE.

Integrity constraints – NOT NULL, CHECK, UNIQUE, PRIMARY KEY, FOREIGN KEY

& Order – (1) where – (2) Group by function cannot be used with where clause

group by

having

③ Joins – Combine the data spread across the table.

Types of Joins –

(1) Simple Join – having a common column.

↳ Equi Join (=)

↳ Non-equi Join (<, >=, <=, >) between ... and ...

(2) Natural Join – All attributes of both table with all tuples.

(3) Cross Join – Cartesian product of two tables.

(4) Outer Join –

↳ left outer join – common tuples with all left table tuples

↳ Right outer join – common tuples with all right table tuples.

↳ Full outer join – All tuples of left table and right table.

④ Subqueries – Nesting of queries one with another

⑤ Indexing – select * from emp e where e.empid = 10

Indexing makes search faster. ↑ Indexing.

Triggers – Never accept any arguments neither they return any value.

Automatically called/fired upon some events.

Assertions – It is a predicate expressing a condition that we wish the database always to satisfy. create assertion <assertion-name> check <predicate>.

Union Capability – Number of columns and data types should be same.

⑥ Relational Algebra –

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

Fundamental operations in the relational algebra are –

(1) select – select tuples that satisfy a given condition. Represented by σ

$\sigma_{branch = "clerk"}(emp)$

(2) Project – select columns that satisfy a given condition. Represented by Π. $\Pi_{cname, sal}(emp)$.

(3) Union – combined two tables, $\Pi_x(A) \cup \Pi_y(B)$.

(4) Set difference – All the tuples in relation (R1) except those tuple that are in R1 not in other relation (R2). $\Pi_x(A) - \Pi_y(B)$.

(5) Set intersection – common tuples in two relations. $\Pi_x(A) \cap \Pi_y(B)$.

(6) Cartesian product – Cross product of two relations, $R_1 \times R_2$.

(7) Rename – Give names to result of relations algebra expressions.

E_1 : Branch = 'clerk' , $\rho_{E_3}(E_1)$

E_2 : amount > 1200 , $\rho_{E_4}(E_2)$ (new name)

$\sigma_{amount > 1200 \wedge Branch = 'clerk'}(emp) \Rightarrow \sigma_{E_3 \cap E_4}$

(8) Natural Join – $\Pi_{cname, loanno, amt}(borrower \bowtie loan)$

(9) Division – "for all" phrase, Denoted by $\div$

$\Pi_{cname, bname}(depositor \bowtie account) \div \Pi_{bname}(\sigma_{city = "Delhi"}(branch))$

(10) Outer Join –

$R_1 \bowtie R_2$ (left outer join)

$R_1 \bowtie R_2$ (Right outer join)

$R_1 \bowtie R_2$ (Full outer join)

(7) Relational Calculus -

It is a non-procedural language. A query is expressed as a formula consisting of number of variables and the expressions involving this variables. No mechanism to specify how the formula should be evaluated.

Types of relational calculus -

(1) Tuple Relation Calculus -

It is a non-procedural query language. It describes the desired information without giving a specific procedure for obtaining that information.

A query in the tuple relational calculus is expressed as

$$\{t \mid P(t)\}$$

t is the set of all tuples such that predicate P is true for t.

(2) Domain Relation Calculus -

It uses domain variables that take on values from an attributes domain, rather than values for an entire tuple.

An expression in the domain relational calculus is of the form

$$\{\langle x_1, x_2, \ldots, x_n \rangle \mid P(x_1, x_2, \ldots, x_n)\}$$

where $x_1, x_2, \ldots, x_n$ represent domain variables

P represent formula composed of atoms.

(8) Atoms that build the formula are -

(1) Set of attributes and constants

(2) Set of comparison operators :- $<, \leq, =, \neq, \geq, >$

(3) Set of connectives :- AND ($\wedge$), OR ($\vee$), NOT ($\sim$)

(4) Implication ($\Rightarrow$) :- $x \Rightarrow y$, if x is true then y is also true.

(5) Set of quantifiers :-

→ $\exists t \in r\,(Q(t)) \equiv$ "there exists" a tuple in t in relation r such that predicate Q(t) is true.

→ $\forall t \in r\,(Q(t)) \equiv$ Q(t) is true "for all" tuples t in relation r.