Unit 4

The relational data model and relational database constraints

Relational model concepts

Domain :- The set of all permitted values for an attribute is called it's domain. In relational database a domain for an attribute is defined by data types and other constraints such as Not null, auto_inc etc. for example; domain for the age attribute may be the set of all integers between 15 to 50, the domain for the GPA attribute may be the set of all floating point numbers between 0 to 4.

Attributes :- The additional describing characteristics of an entity are called attributes. In relational model, attributes are the columns in a relation/table. In relational model, attributes are used to represent the schema. For eg Student (stud-id, name, add, gender, cont), is a relational schema with attributes stud-id, name etc. The degree (or arity) of a relation is the no. of attributes used in it's schema.

Relation Schema :- A relation schema R denoted by R (A1, A2, .., An), consists of the relation name R and a no. of attributes A1, .., An. A relation schema represents the design of the table. Designing a relational database means designing a no. of relation schema and defining the association between them.

Relation:- A relation r of the relation schema $R(A_1, A_2, \cdots, A_n)$ also denoted by $r(R)$ is a set of n-tuples $r = \{t_1, t_2, \cdots, t_m\}$. Each n-tuple t is an ordered list of n values $t = \langle v_1, v_2, \cdots, v_n \rangle$ where each value v_i is an element of domain $D(A_i)$ or NULL. The ith value in tuple t corresponding to attribute A_i is denoted by $t[A_i]$ or $t.A_i$.

Eg: Student (Sid, name, add, contact) is a relation schema. The relation (or relation state) for this is a table shown below.

1	Ram	PKR	123
2	Hari	Ktm	456
3	Rita	PKR	789

← relation state of schema student.

Each item in a relation is called as tuple. A tuple is a row in a relation. A relation schema is also termed as relation intension and a relation state is also termed as relation extension. A relation schema is a table definition whereas a relation state is the set of records in the table.

Tuple:- A single row or record in a table is called as tuple. A tuple represents a single item in a table. A set of tuples constitute a relation.

An n-tuple is a tuple containing the values for n attributes.

Characteristics of relations

(i) Ordering of tuples in a relation

A relation is an abstract data model. A set of tuples constitute a relation. The ordering of tuples is not a part of relation definition. A relation can have a no. of ordering of tuples. This characteristic of relations provides a no. of flexibilities to the relation. We can view a single relation in multiple views based on preference.

(ii) Ordering of values within a tuple

In a relation, ordering of values/Attributes is not an important as long as the correspondence between the attributes is maintained. Without considering the order, we can define the tuple as a mapping from attributes to its domain.

(iii) Values and NULLS in the tuple

It is preferred that each value in a tuple is an atomic value. The composite and multivalued attributes are not allowed. This model is often called as flat relational model. The relation following this characteristic is said to be in first normal form.

A special value called null value is used to represent that the value of an attribute is unknown or is not applicable to the tuple. For eg. contact no. of office may not be applied for some person as the person may be unemployed, residence contact no. may be unknown in both the cases we can use NULL value.

We can have the following two meanings for NULL value.

- Value is unknown
- Value does not exist
- Value undefined

During database design, it is recomended to avoid NULL values as far as possible. The logical operations between NULL values leads to ambiguity. for eg if the address of two students is NULL, it doesn't mean they have the same address.

(iv) Interpretation (meaning) of a relation

A relation schema can be interpreted as a declaration. The meaning of a relation can be obtained by combining the relation schema and the tuples. for eg: STUDENT (sid, Name, add, contact) is a relation schema, the tuple "1, Ram, Pokhara, 123" gives the meaning with respect to the above schema.

A relation doesn't always represent fact about an entity, it can also represent fact about a relationship. for eg. a relation schema STD_DEPART (sid, dipart) represents a relation between the entities student & department.

Relational model notation

- A relation schema R of degree n is denoted by $R(A_1, A_2, \ldots, A_n)$.
- The uppercase letters denote relation names. for eg: P, Q, R etc.
- The lowercase letters p, q, r denote relation state. The letters t, u, v denote the tuples.
- The name of a relation schema can be used to indicate the current set of tuples (or the relation state). Eg: STUDENT indicates the current relation state, whereas STUDENT (sid, name, ...) represents the relation schema.
- An attribute A of a relation R is denoted by using the dot notation ~~such~~ as R.A. Thus we can use same name for two attributes participating in two different relations.
- An n-tuple t in a relation r(R) is denoted by $t = \langle v_1, v_2, \ldots, v_n \rangle$, where v_i is the value corresponding to attribute A_i.
- We use the notation t[Ai], or t.Ai or t[i] to refer to the value v_i of attribute Ai in tuple t.

- We use the notation $t[A_u, A_w, \ldots, A_z]$ or $t.(A_u, A_w, A_z)$ to refer to the ~~va~~ sub-tuple of values $\langle v_u, v_w, \ldots, v_z \rangle$ from the tuple t.

Relational model constraints & relational database schemas :-

A database contains a no. of relations. The state of a database depends on the states of all its relations at a particular point in time. The states of relations depends on various constraints. These constraints are derived from the rules in the mini-world. The constraints on database depends on the miniworld conditions.

Constraints on relational model can be divided into three categories.

(i) Constraints that are inherent in the data model. They are also called inherent model-based constraints or implicit constraints.

(ii) Constraints that are directly expressed in the schemas of the data model. They are specified in the DDL while constructing the schema. They are also called schema-based constraints or explicit Constraints.

(iii) Constraints that cannot be directly expressed in the schemas of the database and hence must be expressed & enforced by the applicatio

program ~~and~~ or some other ways. They are also called application-based or semantic constraints or business rules.

For example, the constraints that a relation cannot have duplicate tuples is an inherent constraint. The constraints we implement to schema using DDL such as NULL value constraints, data type constraints, auto increment constraints etc are the schema-based constraints. Data validation constraints are the application based constraints.

following are the common constraints implemented in the database in relational-model.

1. Domain Constraints:

The domain constraint specifies that within each tuple, the value of each attribute must be an atomic value from the domain of that attribute. The domain of an attribute depends mainly on the data type selected for the attribute. Domain constraints are mainly the schema-based constraints. Domain constraints are implemented using DDL while designing the schema.

2. Key Constraints and constraints on NULL values:

A relation is defined as the set of ~~tuples~~ distinct tuples. A subset of relation schema R is said to be a super-key if no two tuples in the relation

have the same values for these attributes. Let SK be one such subset of attributes in relation R. Then SK is called as the super key if $t_1[SK] \neq t_2[SK]$.

Every relation has one default super key which is the set of all its attributes.

A minimal superkey is called as the key of the relation. If we remove any attribute from the key, it looses the property of super key. Thus we can say that a key is also a super key but the reverse may not be true always. The key of a relation schema must satisfy the following two properties.

(i) Uniqueness property :- Two distinct tuples in a relation cannot have identical values for the key attributes.

(ii) Minimality property :- If we remove any attr. from the key, it no more remains a super key.

For example: Consider a relation schema
STUDENT (Sid, name, address, contact)
Here, sid is a minimal super key hence it is the key of the schema. Any set of attribute that includes sid is a super key.

We can uniquely identify each tuple in a relation by the values of key attributes. A key may also contain more than one attribute. Every relation schema must satisfy the key constraint. Key of a relation schema is determined from the miniworld.

A relation schema may have more than one key. In this case each key is called as candidate key. For eg: in the schema PERSON (citizen_no, pan_no, name, address, contact), citizen_no & pan_no both are the candidate key.

The candidate key selected to uniquely identify each tuple in a relation is called as primary key of the relation. The primary key attributes are denoted by underlining in a relation. While selecting a primary key we select such

Relational database & relational database schemas:

A relational database schema is a set of relational schemas $S = \{R_1, R_2, \cdots, R_m\}$ and a set of integrity constraints applied between the relations. A relational db state ~~(or relational database)~~ instance is a set of relation states $DB = \{r_1, r_2, \cdots, r_m\}$ where each r_i is a relation state of R_i and each r_i satisfies the integrity constraints defined by the schema.

A relational database is the combination of both relational state and relational schema.

for eg: the relational database schema for the library database may be as shown below.

LIBRARY = { USER, BOOK, ISSUE }
USER = { uid, name, address, contact }
BOOK = { bid, title, author, publisher }
ISSUE = { uid, bid, date_issue, date_return }

The relational state for the above schema may be as follows:

LIBRARY.USER

uid	name	address	contact

LIBRARY.~~USER~~ BOOK

bid	title	author	publisher

ISSUE

bid	uid	date_issue	date_return

Entity Integrity, Referential Integrity & Foreign Key

1. Entity Integrity :- The entity integrity constraint states that value of the primary key attribute cannot be NULL or blank. Primary key value is used to identify each record in a relation. If ~~the value~~ a record has NULL in primary key field, we cannot identify that record in the database.

2. Referential Integrity :- The referential integrity constraint is a constraint that is specified between two relations & is used to maintain the consistency among the tuples in two relations.

   Referential integrity between the relations is maintained by using the foreign key. Informally, referential integrity states that if a tuple in one relation refers to a tuple in another relation, it must refer an existing tuple.

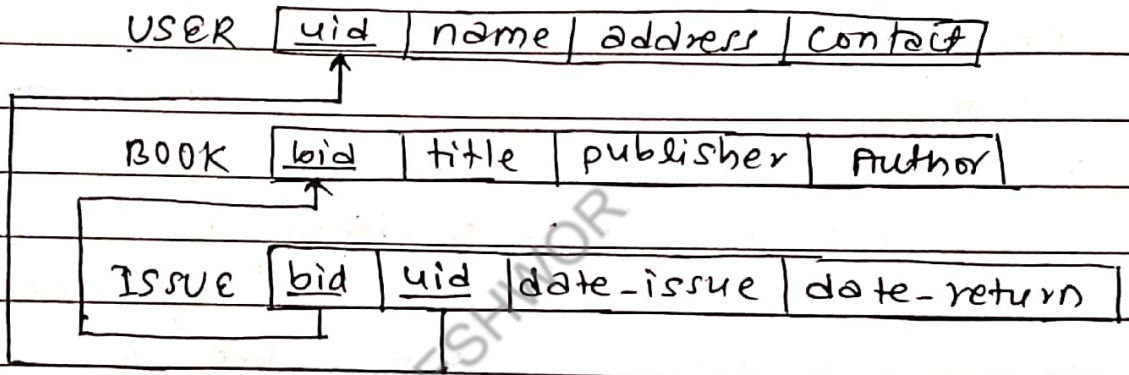
3. Foreign key :- foreign key constraint is a way to implement referential integrity in a relational database. A set of attributes FK in relation schema R1 is said to be a foreign key of R1 referencing the relation R2 if it satisfies the following rules;

   - The attributes of FK have the same domain as the primary key attributes PK of R2.
   - The value of any tuple t_1 for FK in R1 is equal to the value of any tuple t_2 for PK in R2 or NULL.

ie. if t_1 be an arbitrary tuple in R1 & t_2 be an arbitrary tuple in R2, then $t_1[FK] = t_2[PK]$ or NULL

In above definition of foreign key, R1 is called a referencing relation & R2 is called as referenced relation. Defining a foreign key means defining the referencial integrity between two relations.

Eg: Consider the LIBRARY database:



Update operations, Transactions & Dealing with Constraint Violations:

There are three basic operations that update the db.

1. Insert operation:- The insert operation provides a list of attribute values for a new tuple t that is to be inserted into a relation R.

   Insert operation can violate any of the four types of constraints.

- Domain constraint can be violated if an attribute value is given that doesn't belong to the domain of the corresponding attribute.

- Key constraint can be violated if the value of primary key attribute in the tuple is repeated.

- Entity integrity can be violated if any part of the primary key attribute is given NULL value.

- Referential integrity can be violated if the value of any foreign key in a tuple refers to a tuple that doesn't exist in the referenced relation.

While any kind of constraint violation occurs during insertion operation, the DBMS provides a response with error message saying the reason for rejecting the operation.

2. Delete Operation :-

The delete operation removes one or more tuples from the relation with or without condition.

Delete operation can violate the referential integrity. This occurs if the deleted tuple is being referenced by another tuple in different relation. If the delete operation causes violation, we have a no. of options.

- Reject the delete operation.

- Cascade the delete operation. That means delete the tuple as well as all those tuples referencing the deleted tuple.

- set null or set default. In this approach we delete the tuple and modify the referencing tuples to NULL value or default value. But this is not always possible as it may raise entity integrity violation or key constraint violation.

3. Update Operation :-

The update or modify operation is used to change the values of one or more attributes in one or more tuples of a relation R. Updating an attribute which is neither a part of primary key nor a part of foreign key causes no problem. Constraint violation in such case may be domain constraint. The possible constraint violations are as follows:

- Domain constraint violation may occur if we provide a new value to the attribute from outside of its domain.
- Entity constraint violation may occur if we modify the primary key attribute to NULL value.
- Key constraint violation may occur if we modify the primary key attributes to duplicate values.
- Referential integrity constraint may be violated if we try to modify the values to the foreign key attributes.

DBMS must reject the update operation if it brings domain constraint, entity constraint and key constraint violations. The referential constraint violation may be handled either by rejecting the update operation or by setting NULL value or default value to the referenced tuple.

Transaction :-

A transaction is an executing program that performs some database operations against the query made by application programs. Transactions are responsible for performing any database operation in a DBMS. A transaction should not leave a database in inconsistant state. A transaction may include any no. of database operations such as insert, update, delete etc.