DBMS

(DATABASE MANAGEMENT SYSTEM)

LESSON 2

INTRODUCTION TO RELATIONAL MODEL:

Relational Model was proposed by E.F. Codd to model data in the form of relations or tables. After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages like Oracle SQL, MySQL etc. So we will see what Relational Model is.

What is Relational Model?

Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables). Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE and AGE shown in Table 1.

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

TERMINOLOGIES :

Attribute: Attributes are the properties that define a relation. e.g.; ROLL_NO, NAME

Relation Schema: A relation schema represents name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE and AGE) is relation schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.

Tuple: Each row in the relation is known as tuple. The above relation contains 4 tuples, one of which is shown as:

1 RAM DELHI 9455123451 18

Relation Instance: The set of tuples of a relation at a particular instance of time is called as relation instance. Table 1 shows the relation instance of STUDENT at a particular time. It can change whenever there is insertion, deletion or updation in the database.

Degree: The number of attributes in the relation is known as degree of the relation. The STUDENT relation defined above has degree 5.

Cardinality: The number of tuples in a relation is known as cardinality. The STUDENT relation defined above has cardinality 4.

NULL Values: The value which is not known or unavailable is called NULL value. It is represented by blank space. e.g.; PHONE of STUDENT having ROLL_NO 4 is NULL.

Constraints In Relational Model :

While designing Relational Model, we define some conditions which must hold for data present in database are called Constraints. These constraints are checked before performing any operation (insertion, deletion and updation) in database. If there is a violation in any of constrains, operation will fail.

Domain Constraints: These are attribute level constraints. An attribute can only take values which lie inside the domain range. e.g,; If a constrains AGE>0 is applied on STUDENT relation, inserting negative value of AGE will result in failure.

Key Integrity: Every relation in the database should have atleast one set of attributes which defines a tuple uniquely. Those set of attributes is called key. e.g.; ROLL_NO in STUDENT is a key. No two students can have same roll number. So a key has two properties:

It should be unique for all tuples.
It can’t have NULL values.

Referential Integrity: When one attribute of a relation can only take values from other attribute of same relation or any other relation, it is called referential integrity.

KEYS IN RELATIONAL MODEL :

Candidate Key: The minimal set of attributes that can uniquely identify a tuple is known as a candidate key. For Example, STUD_NO in STUDENT relation.

The value of the Candidate Key is unique and non-null for every tuple.
There can be more than one candidate key in a relation. For Example, STUD_NO is the candidate key for relation STUDENT.
The candidate key can be simple (having only one attribute) or composite as well. For Example, {STUD_NO, COURSE_NO} is a composite candidate key for relation STUDENT_COURSE.
No of candidate keys in a Relation are nC(floor(n/2)),for example if a Relation have 5 attributes i.e. R(A,B,C,D,E) then total no of candidate keys are 5C(floor(5/2))=10.
Note – In SQL Server a unique constraint that has a nullable column, allows the value 'null' in that column only once. That's why the STUD_PHONE attribute is a candidate here, but can not be 'null' values in the primary key attribute.

Super Key: The set of attributes that can uniquely identify a tuple is known as Super Key. For Example, STUD_NO, (STUD_NO, STUD_NAME), etc.

Adding zero or more attributes to the candidate key generates the super key.
A candidate key is a super key but vice versa is not true.

Primary Key: There can be more than one candidate key in relation out of which one can be chosen as the primary key. For Example, STUD_NO, as well as STUD_PHONE both, are candidate keys for relation STUDENT but STUD_NO can be chosen as the primary key (only one out of many candidate keys).

Alternate Key: The candidate key other than the primary key is called an alternate key. For Example, STUD_NO, as well as STUD_PHONE both, are candidate keys for relation STUDENT but STUD_PHONE will be an alternate key (only one out of many candidate keys).

Foreign Key: If an attribute can only take the values which are present as values of some other attribute, it will be a foreign key to the attribute to which it refers. The relation which is being referenced is called referenced relation and the corresponding attribute is called referenced attribute and the relation which refers to the referenced relation is called referencing relation and the corresponding attribute is called referencing attribute. The referenced attribute of the referenced relation should be the primary key for it. For Example, STUD_NO in STUDENT_COURSE is a foreign key to STUD_NO in STUDENT relation.

It may be worth noting that unlike, Primary Key of any given relation, Foreign Key can be NULL as well as may contain duplicate tuples i.e. it need not follow uniqueness constraint.

For Example, STUD_NO in STUDENT_COURSE relation is not unique. It has been repeated for the first and third tuples. However, the STUD_NO in STUDENT relation is a primary key and it needs to be always unique and it cannot be null.

ANOMALIES :

An anomaly is an irregularity, or something which deviates from the expected or normal state. When designing databases, we identify three types of anomalies: Insert, Update and Delete.

Insertion Anomaly in Referencing Relation:

We can't insert a row in REFERENCING RELATION if referencing attribute's value is not present in referenced attribute value. e.g.; Insertion of a student with BRANCH_CODE 'ME' in STUDENT relation will result in error because 'ME' is not present in BRANCH_CODE of BRANCH.

Deletion/ Updation Anomaly in Referenced Relation:

We can't delete or update a row from REFERENCED RELATION if value of REFERENCED ATTRIBUTE is used in value of REFERENCING ATTRIBUTE. e.g; if we try to delete tuple from BRANCH having BRANCH_CODE 'CS', it will result in error because 'CS' is referenced by BRANCH_CODE of STUDENT, but if we try to delete the row from BRANCH with BRANCH_CODE CV, it will be deleted as the value is not been used by referencing relation. It can be handled by following method:

ON DELETE/UPDATE SET NULL: If a tuple is deleted or updated from referenced relation and referenced attribute value is used by referencing attribute in referencing relation, it will delete/update the tuple from referenced relation and set the value of referencing attribute to NULL.

ON DELETE CASCADE: It will delete the tuples from REFERENCING RELATION if value used by REFERENCING ATTRIBUTE is deleted from

REFERENCED RELATION. e.g;, if we delete a row from BRANCH with BRANCH_CODE 'CS', the rows in STUDENT relation with BRANCH_CODE CS (ROLL_NO 1 and 2 in this case) will be deleted.

ON UPDATE CASCADE: It will update the REFERENCING ATTRIBUTE in REFERENCING RELATION if attribute value used by REFERENCING ATTRIBUTE is updated in REFERENCED RELATION. e.g;, if we update a row from BRANCH with BRANCH_CODE 'CS' to 'CSE', the rows in STUDENT relation with BRANCH_CODE CS (ROLL_NO 1 and 2 in this case) will be updated with BRANCH_CODE 'CSE'.

COUNTING SUPER KEYS IN MODEL :

Any subset of attributes of a table that can uniquely identify all the tuples of that table is known as a Super key. It's different from the primary and candidate keys in the sense that only the minimal superkeys are the candidate/primary keys.

This means that from a super key when we remove all the attributes that are unnecessary for its uniqueness, only then it becomes a primary/candidate key. So, in essence, all primary/candidate keys are super keys but not all super keys are primary/candidate keys. By the formal definition of a Relation(Table), we know that the tuples of a relation are all unique. So, the set of all attributes itself is a super key.

Example-1 : Let a Relation R have attributes {a1,a2,a3} and a1 is the candidate key. Then how many super keys are possible?
Here, any superset of a1 is the super key.
Super keys are = {a1, a1 a2, a1 a3, a1 a2 a3}
Thus we see that 4 Super keys are possible in this case.

In general, if we have 'N' attributes with one candidate key then the number of possible superkeys is 2(N – 1).

Example-2 : Let a Relation R have attributes {a1, a2, a3,…,an}. Find Super key of R.
Maximum Super keys = 2n – 1.
If each attribute of relation is candidate key.

Example-3: Let a Relation R have attributes {a1, a2, a3,…, an} and the candidate key is "a1 a2 a3" then the possible number of super keys?
Following the previous formula, we have 3 attributes instead of one. So, here the number of possible super keys is 2(N-3).

Example-4: Let a Relation R have attributes {a1, a2, a3,…, an} and the candidate keys are "a1", "a2" then the possible number of super keys?
This problem now is slightly different since we now have two different candidate keys instead of only one. Tackling problems like these is shown in the diagram below:

$\rightarrow |A1 \cup A2| = |A1| + |A2| - |A1 \cap A2|$
= (super keys possible with candidate key A1) + (super keys possible with candidate key A2) – (common superkeys from both A1 and A2)

= 2(n-1) + 2(n-1) – 2(n-2)

Example-5: Let a Relation R have attributes {a1, a2, a3,…, an} and the candidate keys are “a1”, “a2 a3” then the possible number of super keys?
Super keys of (a1) + Super keys of (a2 a3) – Super keys of (a1 a2 a3)
⇒ 2(n – 1) + 2(n – 2) – 2(n – 3)

Example-6: Let a Relation R have attributes {a1, a2, a3,…, an} and the candidate keys are “a1 a2”, “a3 a4” then the possible number of super keys?
Super keys of(a1 a2) + Super keys of(a3 a4) – Super keys of(a1 a2 a3 a4)
⇒ 2(n – 2) + 2(n – 2) – 2(n – 4)

Example-7: Let a Relation R have attributes {a1, a2, a3,…, an} and the candidate keys are “a1 a2”, “a1 a3” then the possible number of super keys?
Super keys of (a1 a2) + Super keys of (a1 a3) – Super keys of(a1 a2 a3)
⇒ 2(n – 2) + 2(n – 2) – 2(n – 3)

Example-8 : Let a Relation R have attributes {a1, a2, a3,…,an} and the candidate keys are "a1", "a2", "a3" then the possible number of super keys? In this question, we have 3 different candidate keys. Tackling problems like these are shown in the diagram below.

→ $|A1 \cup A2 \cup A3| = |A1| + |A2| + |A3| - |A1 \cap A2| - |A1 \cap A3| - |A2 \cap A3| + |A1 \cap A2 \cap A3|$

= (super keys possible with candidate key A1) + (super keys possible with candidate key A2) + (super keys possible with candidate key A3) – (common super keys from both A1 and A2) – (common super keys from both A1 and A3) – (common super keys from both A2 and A3) + (common super keys from both A1, A2, and A3)

$= 2(n-1) + 2(n-1) + 2(n-1) - 2(n-2) - 2(n-2) - 2(n-2) + 2(n-3)$