CHAPTER 4

THE RELATIONAL DATA MODEL AND RELATIONAL DATABASE CONSTRAINTS

LEARNING OBJECTIVES

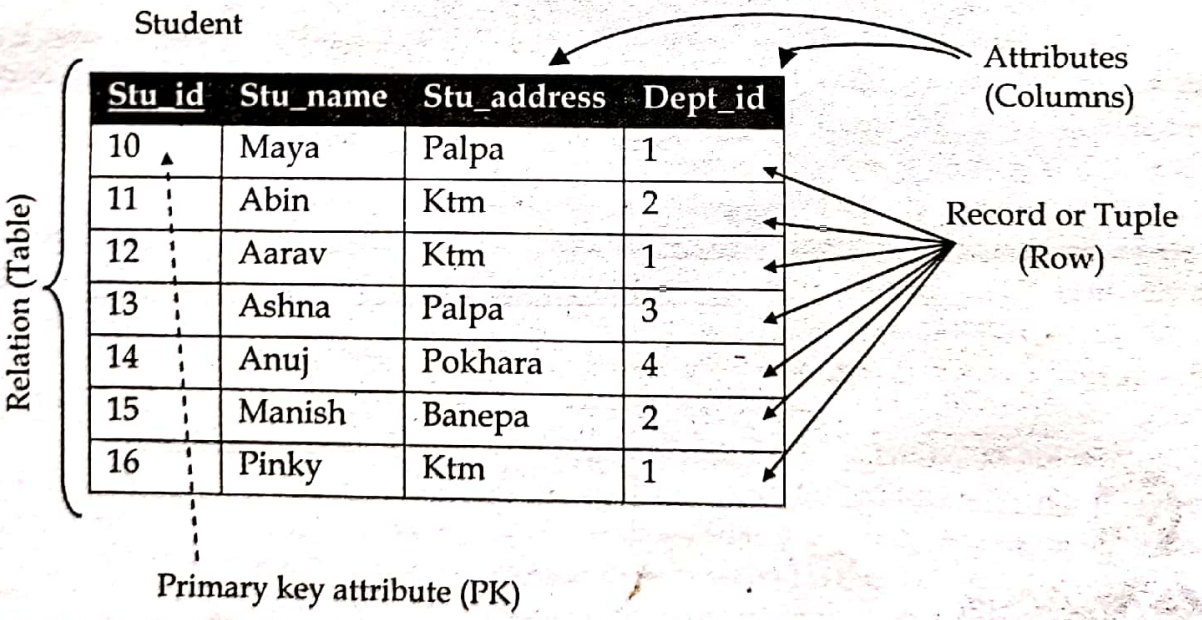
After comprehensive study of this Chapter, you will be able to:

- Relational Model Concepts
- Relational Model Constraints and Relational Database Schemas
- Update Operations, Transactions, and Dealing with Constraint Violations

4.1 Relational Model Concepts

In relational model, the data and relationships are represented by collection of inter-related tables. Each table is a group of column and rows, where column represents attribute of an entity and rows represents records. Relational data model represents the logical view of how data is stored in the relational databases. There exist some concepts related to this, which includes the following terms.

- **Table:** In relational data model, data is stored in the tables. The table consists of a number of rows and columns. Thus, table is used because it can represent the data in the simplest form possible making data retrieval very easy.
- **Attribute:** Any relation have definite properties that are called as attributes.
- **Tuple:** Rows of table represents the tuple which contains the data records.
- **Domain:** Domain is a set of values which is indivisible i.e. value for each attribute present in the table contains some specific domain in which the value needs to lie. For Example: The value of date of birth must be greater than zero. As, it cannot be negative. This is called domain of an attribute.
- **Relation:** A relation in relational data model represents the respective attributes and the correlation among them.



Stu_id	Stu_name	Stu_address	Dept_id
10	Maya	Palpa	1
11	Abin	Ktm	2
12	Aarav	Ktm	1
13	Ashna	Palpa	3
14	Anuj	Pokhara	4
15	Manish	Banepa	2
16	Pinky	Ktm	1

4.2 Structure of Relational Databases Schemas & Instances

The relational model represents both data and the relationships among those data using relation. A relation is used to represent information about any entity (such as student, staff, Department, subject etc.) and its relationship with other entities in the form of attributes (or columns) and tuples (or rows). A relation comprises of a relation schema and a relation instance. A relation schema (also termed as relation intension) depicts the attributes of the table and a relation instance (also termed as relation extension) is a two-dimensional table with a time-varying set of tuples.

A relation schema consists of a relation name R and a set of attributes (or fields) A1, A2, ..., An. It is represented by R (A1, A2... An) and is used to describe a relation R. A set of relation schemas {R1, R2..... Rm} together with a set of integrity constraints in the database constitutes relational database schema. For example, the relational database schema CMS is a set of relation schemas, namely, Student, Staff, Subject, Marks, and College, which is shown below:

College Management System (CMS)

Department (Dept_id, Dept_name, Dept_block_no)

Student (Stu_id, Stu_name, Stu_address, Dept_id)

Staff (Staff_id, staff_name, Dept_id)

Subject (Sub_id, Sub_name, Sub_code, staff_id)

Marks (Marks_Obtain, Sub_id, Stu_id)

For example, a relational database instance corresponding to the **Student** schema is shown below:

Department

Dept_id	Dept_name	Dept_block_no
1	Computer	100
2	Mathematics	200
3	Economics	300
4	Account	400
5	Physics	500

Staff

Staff_id	Staff_name	Dept_id
11	Mohan	1
22	Pratima	2
33	Madan	1
44	Kamala	3
55	Sandhya	4
66	Umesh	3
77	Ramesh	1

Student

Stu_id	Stu_name	Stu_address	Dept_id
10	Maya	Palpa	1
11	Abin	Ktm	2
12	Aarav	Ktm	1
13	Ashna	Palpa	3
14	Anuj	Pokhara	4
15	Manish	Banepa	2
16	Pinky	Ktm	1

Marks

Marks_Obtain	Sub_id	Stu_id
60	20	10
55	21	12
58	22	10
49	20	13
61	23	14
67	22	13
60	24	16
55	26	15
33	25	11

Subject

Sub_id	Sub_name	Sub_code	Staff_id
20	DBMS	D-20	11
21	C++	C-21	22
22	NM	N-22	11
23	TOC	T-23	77
24	PHP	P-24	44
25	AI	A-25	33
26	ASP	A-26	55
27	CG	C-27	66
28	C-Prog	C-28	44

4.3 Relational Database Schemas

A relational database schema helps you to organize and understand the structure of a database. This is particularly useful when designing a new database, modifying an existing database to support more functionality, or building integration between databases.

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema's underlying mission business rules for which the database is created. In a schema diagram, all database tables are designated with unique columns and special features, e.g., primary keys, foreign keys or not null, etc. Formats and symbols for expression are universally understood, eliminating the possibility of confusion. The table relationships also are expressed via a parent table's primary key lines when joined with the child table's corresponding foreign keys. Schema diagrams have an important function because they force database developers to transpose ideas to paper. This provides an overview of the entire database, while facilitating future database administrator work.

Example: College Management System (CMS)

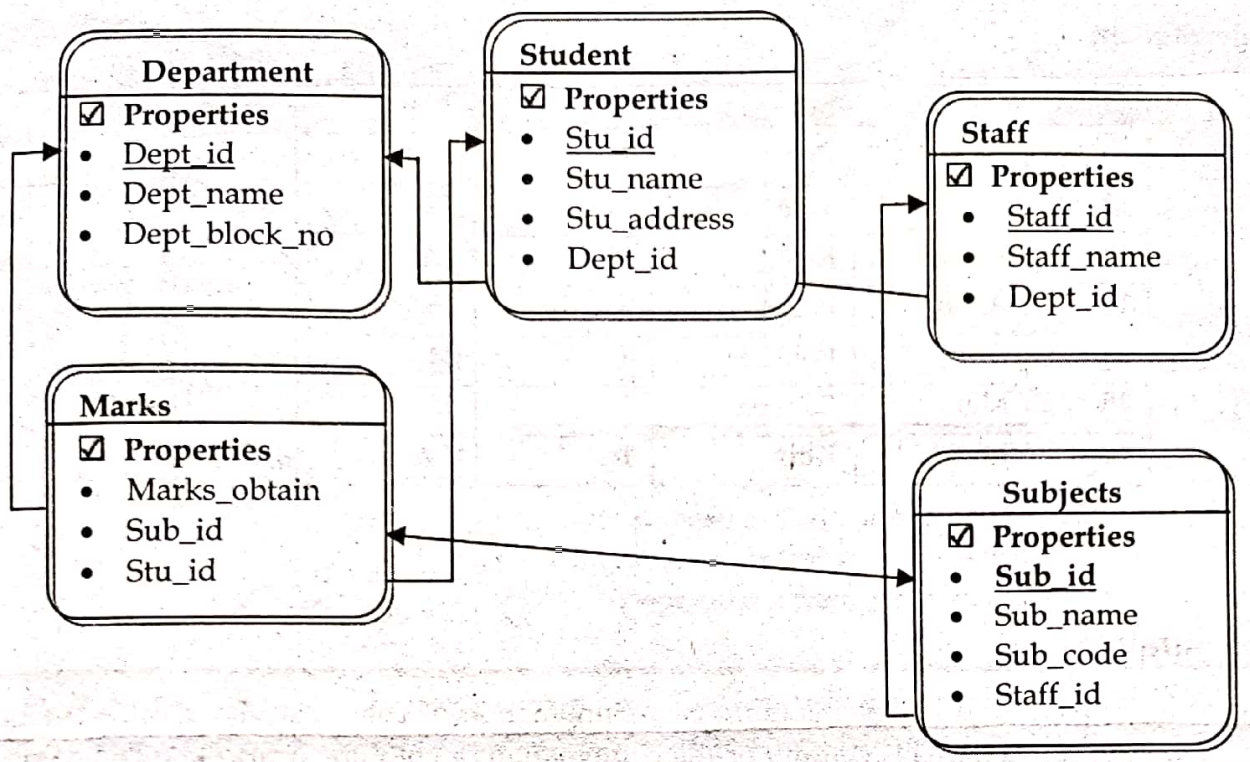
Department (Dept_id, Dept_name, Dept_block_no)

Student (Stu_id, Stu_name, Stu_address, Dept_id)

Staff (Staff_id, staff_name, Dept_id)

Subject (Sub_id, Sub_name, Sub_code, staff_id)

Marks (Marks_Obtain, Sub_id, Stu_id)



4.4 Relational Model Constraints

Relational Integrity constraints are referred to conditions which must be present for a valid relation. These integrity constraints are derived from the rules in the mini-world that the database represents. There are many types of integrity constraints. A constraint on the Relational database management system is mostly divided into three main categories are:

- Domain constraints
- Key constraints
- Referential integrity constraints.

Domain Constraints

Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type. A table in DBMS is a set of rows and columns that contain data. Columns in table have a unique name, often referred as attributes in DBMS. A domain is a unique set of values permitted for an attribute in a table. For example, a domain of month-of-year can accept January, February....December as possible values, a domain of integers can accept whole numbers that are negative, positive and zero.

Domain constraints can be defined as the definition of a valid set of values for an attribute. The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.

Example: Student

Stu_id	Stu_name	Stu_address	Dept_id	Age
10	Maya	Palpa	1	22
11	Abin	Ktm	2	32
12	Aarav	Ktm	1	43
13	Ashna	Palpa	3	55
14	Anuj	Pokhara	4	34
15	Manish	Banepa	2	32
16	NULL	Ktm	1	AA

Not allowed because age is integer.

Key constraints

A primary key constraint declares a column or a combination of columns whose values uniquely identify each row in a table. This column or the combination of columns is also known as primary key of the table. If we insert or update a row that would cause duplicate primary key, database will issue an error message. In other words, a primary key constraint helps enforce the integrity of data automatically.

The Key Constraint specifies that there should be such an attribute in a table, which can be used to fetch data for any tuple. The Key attribute should never be NULL or same for two different rows of data.

Example: In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =" Google".

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
1	Gmail	Active

Key constraint violets

- Must be one* key attribute
- key attr. value must not be NULL.
- Key attribute value should not repeate.

Referential integrity constraints

A referential integrity constraint is specified between two tables. In the referential integrity constraints, if a foreign key in table 1 refers to the primary key of table 2, then every value of the foreign key in table 1 must be null or be available in table 2.

Student

Stu_id	Stu_name	Dept_id
10	Maya	1
11	Abin	2
12	Aarav	1
13	Ashna	3
14	Anuj	4
15	Manish	2
16	Pinky	6

Department

Dept_id	Dept_name	Dept_block_no
1	Computer	100
2	Mathematics	200
3	Economics	300
4	Account	400
5	Physics	500

Not allowed as Dept_id 6 is not defined as Primary key of table Department, Dept_no is a foreign key defined

4.5 Update Operations, Transactions, and Dealing with Constraint Violations

There are three basic operations that can change the states of relations in the database. Insert, delete, and update (or modify). The insert new data, delete old data, or modify existing data records. Insert is used to insert one or more new tuples in a relation, delete is used to delete tuples and update is used to change the values of some attributes in existing tuples. Whenever these operations are applied, the integrity constraints specified on the relational database schema should not be violated. In this section we discuss the types of constraints that may be violated by each of these operations and the types of actions that may be taken if an operation causes a violation.

The 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. Domain constraints can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type. Key constraints can be violated if a key value in the new tuple t already exists in another tuple in the relation R. Referential integrity can be violated if the value of any foreign key in t refers to a tuple that does not exist in the referenced relation.

Example

Department

Dept_id	Dept_name	Dept_block_no
1	Computer	100
2	Mathematics	200
3	Economics	300
4	Account	400
5	Physics	500

Staff

Staff_id	Staff_name	Dept_id
11	Mohan	1
22	Pratima	2
33	Madan	1
44	Kamala	3
55	Sandhya	4
66	Umesh	3
77	Ramesh	1

Here in the staff table we cannot insert new record {6, "English", 454.50} because value of attribute Dept_block_no is in floating point which violets the domain constraint.

Similarly, we cannot insert a new record {4, "English", 700} to department table because the key value 4 already exist in the Department table.

Also we cannot insert the new record {12, "Lalit", 6} to Staff table because their reference was not present at Department table.

The delete operation

The delete operation can violate only referential integrity. This occurs if the tuple being deleted is referenced by foreign keys from other tuples in the database. To specify deletion, a condition on the attributes of the relation selects the tuple to be deleted.

From the above tables we cannot delete the record {1, "Computer", 100} from Department table because their reference is saved to their child table Staff so by deleting this it violets the foreign key constraint.

The update operation

The update or modify operation is used to change the values of one or more attributes in a tuple of some relation R. It is necessary to specify a condition on the attributes of the relation to select the tuple to be modified. Here are similar issues as with Insert/Delete operations.

Examples

Here we cannot change the record {1, "Computer", 100} to {9,"Computer", 100} because their reference saved to their child table "Staff".

The Transaction Concept

A database application program running against a relational database typically executes one or more transactions. A transaction is an executing program that includes some database operations, such as reading from the database, or applying instructions, deletions, or updates to the database. At the end of the transaction, it must leave the database in a valid or consistent state that satisfies all the constraints specified on the database schema. A single transaction may involve any number of retrieval operations and any number of update operations. These retrievals and updates will together from an atomic unit of work against the database. For example, a transaction to apply a bank withdrawal will typically read the user account record, check if there is a sufficient balance, and then update the record by the withdrawal amount.

A large number of commercial applications running against relational database in online transaction processing (OLTP) systems are executing transactions at rates that reach several hundred per second. Transaction processing concepts and concurrent execution of transactions and recovery from failures will be discussed in next chapters.

4.6 Advantages of using Relational model

- **Simplicity:** A relational data model is simpler than the hierarchical and network model.
- **Structural Independence:** The relational database is only concerned with data and not with a structure. This can improve the performance of the model.

- **Easy to use:** The relational model is easy as tables consisting of rows and columns is quite natural and simple to understand
- **Query capability:** It makes possible for a high-level query language like SQL to avoid complex database navigation.
- **Data independence:** The structure of a database can be changed without having to change any application.
- **Scalable:** Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.

4.7 Disadvantages of using Relational model

There are a lot of advantages of relational model but some disadvantages of relational model are listed below:

- Few relational databases have limits on field lengths which can't be exceeded.
- Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.
- Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.