UNIT - 4

The Relational data model & Relational Database Constraint

¥ Relational Model Concepts:

Relational model represents the database as a collection of relations (ie.) table of values

→ Every row in the table represents a collection of related data values. The table name & column name are helpful to interpret the meaning of values in each row. The data are represented as a set of relations.

- Table: In relational data model, data is stored in the tables. The table consists of a number of rows and columns. Thus, it is used because it can represent the data in the simplest form possible making data retrieval very easy.

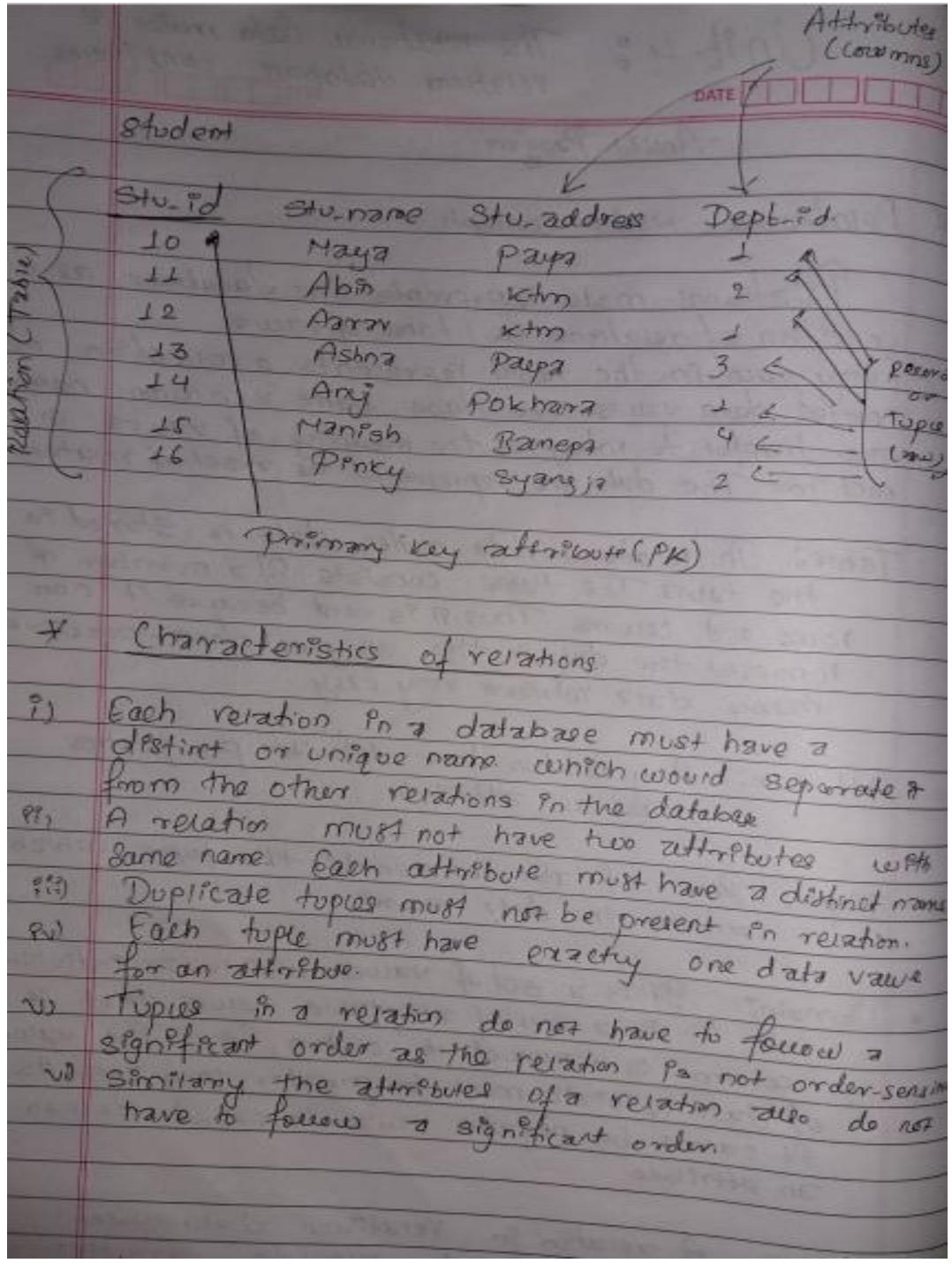
- Attribute: Any relation have definite properties called as attributes.

- Tuple: Rows of table represents the tuple which contains the data records.

- Domain: It is a set of values which is indivisible. ie. It is a set of acceptable values that a column is allowed to contain. ex: 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 & correlation between them.

classmate PAGE



* Characteristics of relations

i) Each relation in a database must have a distinct or unique name which would separate it from the other relations in the database

ii) A relation must not have two attributes with same name. Each attribute must have a distinct name

iii) Duplicate tuples must not be present in relation.

iv) Each tuple must have exactly one data value for an attribute.

v) Tuples in a relation do not have to follow a significant order as the relation is not order-sensitive

vi) Similarly the attributes of a relation also do not have to follow a significant order.

* Relational Model Notation:

We will use following notation in our presentation

i) A relation schema R of degree n is denoted by $R(A_1, A_2, \ldots, A_n)$

ii) The uppercase letters Q, R, S denote relation names

iii) The lowercase letters q, r, s denote relation states

iv) The letters t, u, v denotes tuples

v) The name of relation schema such as STUDENT indicates current set of tuples in that relation whereas STUDENT (Name, Roll, ...) refers only to relation schema.

vi) An attribute A can be qualified with the relation name R to which it belongs by using the dot notation R.A → for example: STUDENT.Name, STUDENT.Age etc

vii) A 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.

→ Both $t[A_i]$ and $t.A_i$ [and sometimes $t[i]$] refer to the value v_i in t for attribute A_i.

→ Both $t[A_u, A_w, \ldots, A_z]$ & $t.(A_u, A_w, \ldots, A_z)$, where $A_u, A_w, \ldots, A_z$ is the list of attributes for R, refer to the subtuple of values $\langle v_u, v_w, \ldots, v_z \rangle$ from t corresponding to the attributes specified in the list

DATE

* Relational Model Constraints and Relational Database Schemas

Relational Model constraints are referred to conditions which must be present for a valid relation. A constraint on the relationship database management system is mostly divided into three main categories:

- Domain Constraints
- Key constraints
- Referential Integrity Constraints

o Domain constraints

→ 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.

→ We perform data type check here, which means when we assign a data type to a column, we limit the values it can contain. Eg: If we assign a datatype of attribute age as int, we can't give it values other than int datatype.

Eg.

Stu_Id	Stu_name	Age
10	Ankit	20
11	Arav	19
12	Anjana	AA ← Not allowed as age is integer.

DATE

Key Constraints

→ A primary key constraint declares a column or a combination of columns whose values uniquely identify each row in a table.

→ If we insert or update a row that would cause duplicate primary key, database will issue an error message.

→ The key constraint specifies that the key attribute should be UNIQUE and should never be NULL. It should not be same for two different rows of data.

Ex: In below table, CustomerID is a key attribute of customer table. It is most likely to have a single key for a single customer, ie. Customer ID: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

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

Example:

Student

Stu_id	stu_name	Dep_id
10	Maya	1
11	Ankit	2
12	Abin	1
13	Aarav	6

Department

Dep_id	Dep_name
1	Computer
2	Math
3	Account
4	Physics

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

DATE

* Relational Database Schemas

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

Formally, a relational database schema is a set of relation schemas $S = \{R1, R2, \ldots, Rm\}$ & a set of integrity constraints. A relational database state of S is a set of relation states $DB = \{r1, r2, \ldots, rm\}$ such that each r_i is a state of R_i and r_i relation states satisfy the integrity constraints.

→ A specific characteristic, that bears the same real-world concept may appear in more than one relationship with the same or a different name. Ex: In Employees relation, Employee Id is represented in vouchers as AuthBy & prepBy

→ The specific real-world concept that appears more than once in a relationship should be represented by different names.

→ The integrity constraints that are specified on database schema shall apply to every database state of that schema.

* Entity Integrity, Relational integrity and Foreign key

Entity integrity

→ The entity integrity states that no primary key value can be NULL. This is because the primary key value is used to identify individual ~~type~~ tuple in the relation. Having NULL values for primary key implies that we cannot identify some tuples ex. If two or more tuples had NULL for their primary keys, then we may not be able to distinguish them if we try to reference them from other relations

Referential Integrity constraint

→ The referential integrity constraint is specified between two relations & is used to maintain the consistency among the tuples in the two relations

→ Referential Integrity constraint typically arise from the relationships among the entities represented by the relation schemas

→ This constraint is compulsory through foreign key. The values of foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take NULL values, but can't be empty

Foreign key

→ It is the field in the table that is primary key of another table. A foreign key may accept multiple NULL values. A foreign key cannot automatically create an index, clustered or non-clustered however, we can manually create an index on foreign key. We have multiple foreign keys in the table

DATE

* Concept of insert, delete and update operations (Dealing with constraint violations)

There are three basic operations that can change the states of relations in the database. Insert, delete or update (modify) whenever these operations are applied, the integrity constants 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 & the types of actions that may be taken if an operation causes a violation

o The insert operation

→ Insert operation is used to insert one or more new tuples in a relation
→ It 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.

DATE

Ex: Department

Dep_id	Dep_name	Dep_block_no
1	Computer	100
2	Math	200
3	Economics	300
4	Account	400
5	Physics	500

Staff

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

→ In staff we cannot add new record {6, "English", 454.50} because value of attribute Dept_block_no is not integer. Which violates the domain constraint

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

→ And we cannot insert the new record. {12, "Ankit", 6} to staff table because their reference was not present at Department table.

The delete operation

-> Delete operation is used to delete tuples in a relation
-> It can violate only referential integrity.
-> This occurs if 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.

In above ex, we cannot delete record [1, "Computer", 100] on department table as their reference is saved to their child table. So, by deleting this it violate foreign key constraint.

The update operation

→ Update operation is used to change the values of some attributes in existing tuples of some relation R.

→ It is necessary to specify a condition on the attributes of the relation to select the tuple to be modified.

Ex: In above ex, we cannot change the record {1, "Computer", 100} to {9, "Computer", 100} because this reference saved to their child table "Staff".

Concept of Transactions.

→ A transaction is an executing program that includes some database operations, such as reading from database, or applying insertions, deletions, or updates to the database.

→ A database application program running against a relational database typically executes one or more transactions.

→ 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.

→ Example: A transaction to apply a bank withdrawal will typically read the user account record, check if there is a sufficient balance, & then update the record by the withdrawal amount.

DATE

* Advantages of using Relational Model

→ It is more simpler than hierarchical and network model.
→ It is concerned with data rather than structure. So this can improve the performance of model.
→ It is easy to use since tables consist of rows and columns is simple to understand.
→ It is data independance since structure of database can be changed without having to change any applicatio
→ It makes possibe for high level query language like SQL to avoid complex database navigatio

* Disadvantages of using Relational Model.

→ Few relational databases have limits on field lengths which can't be exceeded.

→ Relational database can sometimes become complex as the amount of data grows.

→ Complex relational database systems may lead to isolated databases where the information can't be shared from one system to another.

	Security	Integrity
i)	Data security defines the prevention of data corruption through the use of controlled access mechanisms	i) Data Integrity defines the quality of data, which guarantees the data is complete & has a whole structure
ii)	Data security deals with the protection of data.	ii) Data Integrity deals with the validity of data.
iii)	It is making sure only the people who should have access to the data are the only ones who can access the data.	iii) It is making sure the data is correct and not corrupt.
iv)	It refers to making sure that data is accessed by its intended users, thus ensuring the privacy & protection of data.	iv) It refers to the structure of data & how it matches the schema of the database
	Some means of data security: Authentication/ authorization, encryption & masking	v): Backing up data, designing a suitable UI & error/detection/correction data are some means

Questions asked from this chapter

Q. Difference between integrity and security with example. (2076 - 5 marks)

Q. What is integrity? Explain different types of integrity. (2078 - 5 marks)

Q. Define Integrity constraint? Discuss domain constraint with suitable example. (2070 - 5 marks)

Q. Define referential integrity with example. (2074 - 5 mar

Q. Which part of DBMS taken care of the data dictionary? (2073 - 5 marks)

Questions

- What is relational database? Explain different characteristic of a relation. Define domain constraint.
- What are informal design guidelines for relational schemas?
- What is integrity? Explain different types of database integrity.
- Differentiate between Integrity and Security with example.