DBS211 Introduction to Database Systems

WEEK- 2 Relational Data Model

Agenda

- Relational Model
- Keys
- Relationships
- Referential Integrity
- Table Types

Database

- The Database Properties :
  - Centralized repository
  - Real-time
  - Concurrent
  - Interactive (ad-hoc) query

Why DBMSs?

- Some of the advantages of Database Management systems:
  - Data administration
  - Efficient data access
  - Data integrity & security
  - Concurrent access, crash recovery

Data Models

- Data modeling is simply a method of describing data or information.
- It includes:
  - Structure of data
    - In what format data is stored
  - Operations on Data
    - **Selection**
      - Set of queries to fetch the data
    - **Modification**
      - Operations to modify the database and the data
  - Constraints
    - It defines what data can be stored in the database by enforcing rules and limitations

Why Relational Data Model?

- extends the data model by defining relationships between various data points
- eliminate inefficiencies and non-functional dependencies
  - Efficient data access
    - By structuring data
  - Efficient data manipulation
    - By providing set of operations

Relational Data Model Basics

- The relational model represents data as a two dimensional.
- In the relational model, a table is also called a ***relation***.
- In the following table
  - Every row represent a single instance of the entity –ex : department.
  - Every column in a row, represent an attribute or property of that department.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	(null)	1700

Department table:

Attributes

- Columns in a table (relation) are called attributes.
- In the following table (department), the attributes are:
  - Department_id
  - Department_name
  - Manager_id
  - Location_id

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	(null)	1700

Department table:

Schema

- A schema represent the name of a relation and its attributes.

DEPARTMENT(department_id,department_name, manager_id,Location_id)

- A relational database consists of a set of relations.
- A relational database schema is a set of schemas of database relations.

Tuple

- In a relation (table), a row (record) is called a tuple.
- The following relation (department) has 8 tuples.
- The value of each attribute in a tuple represents a component.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	(null)	1700

Department table

Domains

- In a relational model, each components of a tuple must be atomic. Each component is allowed to be a single value of primitive types (integer, character, character string).

- The domain of an attribute (column) is simply the data type of that attribute.

DEPARTMENT(department_id: **int**,department_name: **string**, manager_id : **int**,Location_id: **int**)

Null Value

- Null denotes the absent of a value.
- The value of an attribute is null if
  - It is unknown
  - It does not exist

Integrity Constraints

- Integrity constraints are a set of rules to maintain the quality.
  - Domain/tuple constraints
  - Entity integrity constraints
  - Key constraints
  - Referential integrity constraints

Domain Integrity Constraint

- To retrict the values of tuple attributes in a database relation.
- Domain constraints
  - Define conditions on a single attribute of a tuple

  department_id = 60
- Tuple constraints
  - Define conditions on different attributes of a tuple

  department_id = 60 and location_id =1700

Entity Integrity Constraint

- This constraint ensures that each relation has a primary key.
- A primary key is an attribute or set of attributes that uniquely identifies every tuple in a relation.
- The values of the attributes involved in the primary key cannot be null.
- Attribute department_id is the primary key of relation departments.

  DEPARTMENTS(department_id,department_name, manager_id,Location_id)

  - A primary key is unique and cannot be null.

Referential integrity Constraint

- Referential integrity is a constraint specified on two relations.
- Referential integrity constraint determines if two relations are related. A value appearing in a column of a relation also appears in a column of another relation.

  DEPARTMENTS(department_id,department_name, manager_id,*Location_id*)

- Location ID exits in relations departments and locations.
  - Attribute location_id in relation department refers to the location_id attribute in relations locations.

LOCATIONS(*location_id*,street_address,postal_code,city,state_province,country_id)

Referential Integrity Constraints

Departments

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	(null)	1700

locations

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID
1000	1297 Via Cola di Rie	00989	Roma	(null)	IT
1100	93091 Calle della Testa	10934	Venice	(null)	IT
1200	2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefecture	JP
1300	9450 Kamiya-cho	6823	Hiroshima	(null)	JP
1400	2014 Jabberwocky Rd	26192	Southlake	Texas	US
1500	2011 Interiors Blvd	99236	South San Francisco	California	US
1600	2007 Zagora St	50090	South Brunswick	New Jersey	US
1700	2004 Charade Rd	98199	Seattle	Washington	US
1800	147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA
1900	6092 Boxwood St	YSW 9T2	Whitehorse	Yukon	CA
2000	40-5-12 Laogianggen	190518	Beijing	(null)	CN
2100	1298 Vileparle (E)	490231	Bombay	Maharashtra	IN
2200	12-98 Victoria Street	2901	Sydney	New South Wales	AU
2300	198 Clementi North	540198	Singapore	(null)	SG
2400	8204 Arthur St	(null)	London	(null)	UK
2500	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford	Oxford	UK
2600	9702 Chester Road	09629850293	Stretford	Manchester	UK
2700	Schwanthalerstr. 7031	80925	Munich	Bavaria	DE
2800	Rua Frei Caneca 1360	01307-002	Sao Paulo	Sao Paulo	BR
2900	20 Rue des Corps-Saints	1730	Geneva	Geneve	CH
3000	Murtenstrasse 921	3095	Bern	BE	CH
3100	Pieter Breughelstraat 837	3029SK	Utrecht	Utrecht	NL
3200	Mariano Escobedo 9991	11932	Mexico City	Distrito Federal,	MX

Keys

- A key, in a relation, identifies every tuple uniquely.
  - A key can be a single attribute or a set of attributes.
- In a multi-attribute key, the combinations of key attributes must be unique for each tuple (row) in the relation (table).
  - In relation department, the key is department_id which is unique for every tuple.

DEPARTMENT(department_id: **int**,department_name: **string**, manager_id : **int**,Location_id: **int**)

- To represent key in a relation, key attributes are underlined.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	(null)	1700

Example :
Department table

Which ones can be the keys for this table?

Employees				
First Name	**Last Name**	**Job Title**	**Date of Birth**	**SIN**
Steven	King	President	May 16, 1972	495777777
Trenna	Rajs	Sales Manager	Oct 10, 1983	478333333
Ellen	Abel	Clerk	Feb 21, 1995	123444444
John	Smith	Mechanic	Mar 8, 1992	456555555
Jane	Smith	Receptionist	Aug 17, 1998	788666666

Keys contd..

- Most important functions of a database is the process of querying and manipulating data.
- **a single attribute or a combination of attributes that will uniquely identify a specific instance, or row**, in a table.
- the SIN (Social Insurance Number) is absolutely unique per person and therefore would be a good candidate for uniquely identifying instances of employees.

Types of Keys

Candidate Key	an attribute, or combination of attributes, that could potentially be used to uniquely identify an single instance, or row, in a table
Primary Key	an attribute, or combination of attributes, that has been chosen to uniquely identify a single instance, or row, in a table.
Composite Key	A composite key is the case where multiple attributes make up a candidate or primary key. The uniqueness comes from the unique combination of values. Each single attribute can have repeat values, but there can be no repeats in the combination of values from all attributes that are part of the composite key.
Surrogate Key	Is an artificially added field, or attribute, that is added to replace the existing fields from being the primary key. This occurs most often in 2 cases: 1) when multiple composite fields are chosen, in order to simplify the interaction with the database and 2) when there is not a field, or attribute, available to be a Primary Key
Foreign Key	A foreign key is a constraint applied to a table defining the relationship between two tables

Relationships

- When multiple tables, or entities, are being used, there is often attributes in each table the relates the two tables together.

Players			
PlayerID	**FirstName**	**LastName**	**TeamID**
123	Billl	Marlow	**22**
456	Robert	MacDonald	**22**
78	John	Smith	24

Teams		
TeamID	**TeamName**	**ShortColour**
22	Hornets	Yellow
23	Shooters	Grey
24	Slowpokes	Blue

Relationship Types

- each row is uniquely identified by its' primary key and that values must be atomic, each player can only play on one team.
- In this relationship, the Teams table is known as the Parent Table and the Players Table is the Child Table.
- A foreign key constraint would be created on the child players table referencing the the relationship to the parent tables teamID field.

Find the relationship between Team and Players table

- FOR EACH row in the players table, how many teams can there be? (answer: none or one)
- FOR EACH row in the teams table, how many players can there be? (answer: none, one or more than one)

Three types of relationships:

1-to-many (1-∞)	These relationships are the most common type of relationship and means that in one direction, there can be only one value per row of one the child table, but in the other direction, there can many rows in the child table that reference a single row in the parent table.
1-to-1 (1-1)	These relationships are used to reference multiple attributes of a row, determined by the primary key, where different attributes have varying levels of completeness (i..e they have a value or not). When investigating the relationship from both directions, there can not be more than one related row in either table. An example follows.
many-to-many (∞-∞)	Many-to-Many, also noted M:M or M:N, relationships are a special circumstance where there could be more than one reference to each row of the other table, in both directions. If the league was to decide that a player could play on more than one team, then the design of the tables has to change, but then we would have the scenario where each team could have more than one player and each player could play on more than one team.

Example for relationships

1-to-Many Relationships

•Each gas station can only be affiliated with one oil company (example: Shell, Petro Canada, etc) but each oil company is affiliated with many gas stations

1-to-1 Relationships

•Each employee is a single person and each person can only be a single employee, but might not be an employee

Many-to-Many Relationships

•Each shopper at a store can buy more than one product, and each product can be sold to more than one shopper.

Referential Integrity

Players			
PlayerID	**FirstName**	**LastName**	**TeamID**
123	Billl	Marlow	**22**
456	Robert	MacDonald	**22**
78	John	Smith	24

Teams		
TeamID	**TeamName**	**ShortColour**
22	Hornets	Yellow
23	Shooters	Grey
24	Slowpokes	Blue

Cascading

- To help automate the process and simplify the steps needed to allow these changes to occur, referential integrity has a feature called cascading.
- the two most common ones are
  - cascade updates
  - cascade deletes.

Cascade Updates

Players			
PlayerID	**FirstName**	**LastName**	**TeamID**
123	Billl	Marlow	**22**
456	Robert	MacDonald	**22**
78	John	Smith	24

Teams		
TeamID	**TeamName**	**ShortColour**
22	Hornets	Yellow
23	Shooters	Grey
24	Slowpokes	Blue

If the primary key of the parent record changes for any reason, then the child records that reference the parent value would be automatically updated to match the change.

Cascade Deletes

- If a parent record is deleted, then all child records that refer to the parent record will also be deleted.

Players			
PlayerID	**FirstName**	**LastName**	**TeamID**
123	Billl	Marlow	22
456	Robert	MacDonald	22
78	John	Smith	24

Teams		
TeamID	**TeamName**	**ShortColour**
22	Hornets	Yellow
23	Shooters	Grey
24	Slowpokes	Blue

Table Types

- Data Table
  - The primary purpose of this table is to store raw data for later querying, analyzing and manipulating. The majority of tables in a database are of this type.
- Examples: students, employees, orders, products

Table Types

- **Lookup Table**
  - The primary purpose of a lookup table is to centralize data to avoid repeated data groups. A common use of a lookup table is to populate a dropdown list on a website or user interface., Often lookup tables are parent tables in relationships.
- Examples: Provinces, Countries, Colours, Manufacturers, Gender, Transmission Type

Table Types

- **Junction/Bridge Table**
  - A third table created between two tables to simulate a many-to-many relationship through two opposite 1-to-many relationships. Junction or Bridge tables can often contain data as well, based on the dependencies between the data and the two parent primary keys.

Table Types

- **Temporary Table**
  - A table that has limited lifetime in a database. Often these table lack referential integrity, are not manually manipulated, break several fundamental database design rules, but has a very distinct purpose in a database. These purposes could include:
- moving data from one table to another or importing data from an external source
- migrating an old database design to a new database design
- the static storage of query results to minimize repeated processing required for complex calculations. (Often used for feeding high traffic websites or mobile applications)

Week-2 Summary

- Relational Model
- Keys
- Relationships
- Referential Integrity
- Table Types

NEXT WEEK

- SQL (Structured Query Language)
- Single Table Queries
- DML (SELECT, source FROM, filtering WHERE, Sorting ORDER BY, Aliases, Wildcards, CRUD (INSERT, UPDATE, & DELETE)
- Fundamental concepts of case sensitivity
- date formats, and internationalization.