Lecture 10
Structured Query Language
Summary from previous lecture


2. Database structure:
   • Tables are known as "Relations"
   • Rows are "Tuples"
   • Columns are "Attributes"

Contents

• SQL description:
  • SQL definition
  • SQL datatypes
• SQL Syntax
  • Relation calculus
  • SQL commands
SQL stands for “Structured Query Language.” The Structured Query Language is a relational database language. By itself, SQL does not make a DBMS. SQL is a medium which is used to communicate to the DBMS.

Some of the features of SQL:
– SQL is a language used to interact with the database
– SQL is a data access language
– SQL is based on relational tuple calculus
– SQL is a standard relational database management language
– SQL is a “nonprocedural” or “declarative” language.
Most DBMS allow SQL to be used in two distinct ways:

- Interactive SQL. SQL commands can be typed at the command line directly. The DBMS interprets and processes the SQL commands immediately, and the results are displayed.

- Programmatic SQL. SQL statements are embedded in a host language such as Java, C, Python etc. The host language provides the necessary looping and branching structures and the interface with the user, while SQL provides the statements to communicate with the DBMS.
Datatypes in SQL

1. **String (text):**
   - CHAR datatype (fixed-length character data): `CHAR (n)`
   - VARCHAR datatype (variable-length character string): `VARCHAR (n)`

2. **Numbers:**
   - INTEGER datatype (default 10 digits): `INTEGER(p)`
   - FLOAT datatype (default mantissa 16): `FLOAT(p)`

3. **DATE Datatype**
SQL commands

SQL commands can be classified into three types:
1. Data Definition Language commands (DDL)
2. Data Manipulation Language commands (DML)
3. Data Control Language commands (DCL)
Create Table Command

Steps in Table Creation
1. Identify datatypes for attributes
2. Identify columns that can and cannot be null
3. Identify columns that must be unique
4. Identify primary key–foreign key mates
5. Determine default values
6. Identify constraints on columns (domain specifications)
7. Create the table
Create Table. Syntax

- **Table definition**

  ```sql
  CREATE TABLE <tablename>
  (column-name1 data-type-1 [constraint],
   column-name2 data-type-2 [constraint],
   column-nameN data-type-N [constraint]);
  ```

- **Table populating**

  ```sql
  INSERT INTO <tablename> VALUES
  (‘&columnname1’, ‘&columnname2’, ‘&columnnameN’ );
  ```
Contents

• SQL description:
  • SQL definition
  • SQL datatypes

• SQL Syntax
  • Relation Calculus
  • SQL commands
Tuple Relational Calculus

List of main operations used to manipulate Relations:

- INSERT
- DELETE
- UPDATE
- SELECT
- JOIN
- UNION
**INSERT command**

INSERT is a unary operation – it operates on a single Relation and adds a Tuple to a Relation.

**INSERT INTO** Relation **VALUES** (‘&attribute1’,...)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jill</td>
<td>D</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>Steve</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jill</td>
<td>D</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>Steve</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>Lars</td>
<td>A</td>
</tr>
</tbody>
</table>
DELETE command

DELETE is a unary operation – it operates on a single Relation and deletes a Tuple fulfilling criteria from a Relation

DELETE tuple FROM Relation WHERE attribute# = x

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jill</td>
<td>D</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>Steve</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>Lars</td>
<td>A</td>
</tr>
</tbody>
</table>

After DELETE:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jill</td>
<td>D</td>
</tr>
<tr>
<td>2</td>
<td>Bob</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>Lars</td>
<td>A</td>
</tr>
</tbody>
</table>
UPDATE command

UPDATE is a unary operation – it operates on a single Relation and modifies an attribute in Tuple fulfilling criteria in a Relation

```
UPDATE Relation SET t.a2=data WHERE t.a1=x
```
SELECT command

SELECT is a unary operation – it operates on a single Relation. The SELECT operation creates a new relation R2 from relation R1. The Tuples in R1 is a subset of R2.

```
SELECT &attribute1, &attribute2 FROM tablename;
```
SELECT command

Extension to SELECT command:

• SELECT * FROM R1 WHERE a1=6;
• SELECT * FROM R1 GROUP BY a1;
• SELECT * FROM R1 ORDER BY a1 (ASC, DESC);
• SELECT * FROM R1 HAVING a2>3;
• SELECT a1,a3 FROM R3 WHERE a2 IN (\textit{value1},\textit{value2});
JOIN command

JOIN is a binary operation – it operates two Relations. The JOIN operation creates a new relation R3 from relations R1 & R2 based on common attributes (keys).

```
SELECT R1.a1, R1.a2, R2.a2 FROM R1 JOIN R2 ON R1.A2=R2.A1
```
UNION command

UNION is a binary operation – it operates on two Relations R1 and R2 and creates a new relation R3 in which each tuple is either in R1, in the R2, or in both R1 and R2. The two relations must have the same attributes.

```
SELECT a1, a2 FROM R1;
UNION
SELECT a1, a2 FROM R2;
```