Relational Database Language SQL

Vu Tuyet Trinh
trinhvt-fit@mail.hut.edu.vn

Department of Information Systems
School of Information Technology and Communication
Hanoi University of Technology

Languages of DBMS

- Data Definition Language (DDL)
  - define the logical schema (relations, views, …) and storage schema stored in a Data Dictionary
- Data Manipulation Language (DML)
  - Manipulative populate schema, update database
  - Retrieval querying content of a database
- Data Control Language (DCL)
  - permissions, access control, ...

Data Manipulation Language

- Structured Query Language (SQL)
- A brief history
  - SQL 1
    - The first standard for SQL defined in 1986
    - adopted as an international by Standards Organisation (ISO) in 1987
  - SQL2
    - revised version of the processor (also called SQL 92).
    - adopted as the formal standard language for defining and manipulating relational database.
  - SQL 3
    - extension with additional features such as user-defined data types, triggers, user-defined functions and other Object Oriented features

SQL Retrieval Statement

```
SELECT[all|distinct]
      {*|{table.*|expr[alias]|view.*}
      [,{table.*|expr[alias]}]...}
FROM table [alias][,table[alias]] ...
[WHERE condition]
[GROUP BY expr [,expr] ...]
[HAVING condition]
[{UNION|UNION ALL|INTERSECT|MINUS}
      SELECT ...]
[ORDER BY {expr|position} [ASC|DESC]
         [,expr|position}[ASC|DESC].
```

Example: University Database

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Takes

SID	SNO
1108	21
1108	23
1108	29
8507	23
8507	29

Enrol

SID	Course
3936	101
1108	113
8507	101

Course

No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
18	Algebra	Maths

Select column(s) from a table

- Syntax
  - **SELECT** ColumnName, ColumnName, ...
    FROM TableName
  - **SELECT** *
    FROM TableName
- Example

```
SELECT  Name
FROM    Student
```

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Name
Robert
Glen
Norman
Mary

Retrieving rows

- Syntax

```
SELECT    ColumnName,ColumnName, ...
FROM      TableName
WHERE     condition_expression;
```

- Example

```
SELECT    *
FROM      Student
WHERE     suburb=''Bundoora'' ;
```

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

Condition Expression

- Comparative operations: =, !=, <, >, <=, >=
- Logic operation: NOT, AND, OR
- Other operation: BETWEEN, IN, LIKE
  - Digital data type
    - attr **BETWEEN** val1 **AND** val2 (⇔ (attr>=val1) and (attr<=val2))
    - attr **IN** (val1, val2, ...) (⇔ (attr=val1) or (attr=val2) or ...)
  - String data type
    - **LIKE**: % instead of one character
      * instead of any characters (string)

Join

- Syntax

```
SELECT   T1.C1,T1.C2,T2.C1,T2.C4, ...
FROM     T1, T2
WHERE    condition_expression
```

- Example

```
SELECT   Id, Name, Suburb,Course
FROM     Student,Enrol
WHERE    Id=SID
```

Using AS keyword

- Syntax

```
SELECT   c1 as name1, c2 as name2
FROM     TableName
```

- Example

```
SELECT   SID , Student.Name as SName,
           Subject.Name as Subject
FROM     Student,Takes, Subject
WHERE    (Id=SID) and (SNO = No)
```

Using AS in FROM clause

- Used for naming variable(s)
- Example

```
SELECT   SID , Stud.Name as SName,
           Sub.Name as Subject
FROM     Student as Stud,Takes,
         Subject as Sub
WHERE    (Id=SID) and (SNO = No)
```

DISTINCT keyword

- Syntax

```
SELECT DISTINCT      <bt1>, <bt2>, …
FROM                 <bang1>,<bang2>, …
```

- Example

```
SELECT DISTINCT      Dept
FROM                 Course
```

Ordering Results

- Syntax

```
SELECT      <bt1>, <bt2>, …
FROM        <bang1>,<bang2>, …
[WHERE      <dieu kien chon>]
ORDER BY <tt1> | <bieu thuc so 1> [ASC | DESC]
```

- Example

```
SELECT      Name
FROM        Student
ORDER BY Name ASC
```

Aggregation Function

- on collections of data values
  - AVG (Average)
  - MIN (minimum)
  - MAX (maximum)
  - SUM
  - COUNT
- Often used with GROUP clause

Grouping Results

- Syntax

```
SELECT      <bt1>, <bt2>, …
FROM        <bang1>,<bang2>, …
[WHERE      <dieu kien chon>]
[GROUP BY   <tt1>, <tt2>, …]
```

- Example

```
SELECT    Suburb, Name
FROM      Student
GROUP BY  Suburb

SELECT    Suburb, Count(Id)
FROM      Student
GROUP BY  Suburb
```

Having Clause for Result Display

- Syntax

```
SELECT    <bt1>, <bt2>, …
FROM      <bang1>,<bang2>, …
[WHERE    <dieu kien chon>]
HAVING    <dieu kien in ket qua>
```

- Example

```
SELECT    Suburb, COUNT(ID)
FROM      Student
GROUP BY  Suburb
HAVING    COUNT(ID) > 3
```

Set Operations

- **UNION**, **MINUS**, **INTERSECT**
- Example

```
SELECT DISTINCT     Subject.Name
FROM                Subject
MINUS
SELECT DISTINCT     Subject.Name
FROM                Student, Takes, Subject
WHERE               Student.Id = Takes.SID and
                    Takes.SNO = Subject.No
```

Nested Sub-queries

- Complete select queries within a where clause of another outer query
- Creates an intermediate result
- No limit to the number of levels of nesting
- Used for
  - Verify a element in a set (**IN**)
  - Set comparison **>ALL, >=ALL, <ALL,<=ALL,=ALL, NOT IN, SOME**
  - Verify table with/without record (**EXISTS** hoặc **NOT EXISTS**)

Example: University Database

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Takes

SID	SNO
1108	21
1108	23
1108	29
8507	23
8507	29

Enrol

SID	Course
3936	101
1108	113
8507	101

Course

No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
18	Algebra	Maths

Example

```
Select   *
From     STUDENT
Where    Id     IN
   (select DISTINCT SID
   From         Takes)
```

Other Functions

- Mathematic functions
  - ABS, SQRT, LOG, EXP, SIGN, ROUND
- String functions
  - LEN, LEFT, RIGHT, MID
- Time functions
  - DATE, DAY, MONTH, YEAR, HOUR, MINUTE, SECOND

Insert Data

- Syntax
  - **INSERT INTO** table[(col1,col2,…)]
    VALUES (exp1,exp2,…)
  - **INSERT INTO** table[(col1,col2,…)]
    SELECT col1,col2, …
    FROM tab1, tab2, …
    WHERE <dieu_kien>
- Example
  - **INSERT INTO** Student(Id, Name, Suburb)
    VALUES (''1179'',''David'',''Evr'')

Update, Delete Data

- Syntax for deleting data

```
DELETE FROM     table
WHERE           cond_exp;
```

- Syntax for updating data

```
UPDATE  table
SET     col1 = exp1,
        col2=exp2,
        col2=exp2,
WHERE   cond_exp;
```

- Example
  - **DELETE FROM** Student
    WHERE Suburb = ‘‘Bundoora’’;
  - **UPDATE** Student
    SET Suburb = ‘‘Evry’’
    WHERE Suburb = ‘‘Evr’’;

Exercises