SQL – Part 3

Data Manipulation Language (Simple SQL Queries)

Simple SQL Queries

- SQL provides the SELECT statement for retrieving data from a database.
- The SELECT statement has the following basic form:

```
   SELECT attribute_list
     FROM table_list
   [WHERE condition]
[GROUP BY attribute_list [HAVING group_condition]]
[ORDER BY attribute_list];
```

Note:

- Only SELECT and FROM are mandatory.
- The symbol $*$ means all the attributes.
- Attribute names may be qualified with the table name (required, if attribute-names are not unique).
- Attribute and table names can be given an alias.
- DISTINCT is used for removing duplicate tuples in the query result.

SQL Queries – Select Clause

ENROL				
<u>StudentID</u>	<u>CourseNo</u>	<u>Semester</u>	Status	EnrolDate
456	COMP2600	2016 S2	active	25/02/2016
458	COMP1130	2016 S1	active	25/02/2016
456	COMP2400	2016 S2	active	09/03/2016

```
SELECT * FROM ENROL;
```

StudentID	CourseNo	Semester	Status	EnrolDate
456	COMP2600	2016 S2	active	25/02/2016
458	COMP1130	2016 S1	active	25/02/2016
456	COMP2400	2016 S2	active	09/03/2016

Australian National University

SQL Queries – Select Clause

ENROL				
StudentID	CourseNo	Semester	Status	EnrolDate
456	COMP2600	2016 S2	active	25/02/2016
458	COMP1130	2016 S1	active	25/02/2016
456	COMP2400	2016 S2	active	09/03/2016

```
SELECT ENROL.StudentID, Semester FROM ENROL;

SELECT e.StudentID as SID, e.Semester FROM ENROL e;

SELECT DISTINCT StudentID, Semester FROM ENROL;
```

StudentID	Semester
456	2016 S2
458	2016 S1
456	2016 S2

SID	Semester
456	2016 S2
458	2016 S1
456	2016 S2

StudentID	Semester
456	2016 S2
458	2016 S1

SQL Queries – Where Clause

- Unspecified `WHERE`-clause means no condition.
  - all tuples of a relation in the `FROM`-clause are selected.
  - if multiple relations are specified in the `FROM`-clause without join conditions, the Cartesian product of relations is selected (**be careful**).
- The condition in the `WHERE`-clause can be simple or complicated.

```
SELECT * FROM STUDENT;
SELECT * FROM STUDENT, COURSE;
SELECT * FROM STUDENT WHERE StudentID BETWEEN 100 AND 500;
SELECT * FROM STUDENT WHERE Email is NOT NULL;
SELECT * FROM STUDENT WHERE Email like '%@gmail.com';
```

- **Question:** *Assume that we have 1000 tuples in* STUDENT *and 100 tuples in* COURSE*. How many tuples we will have in the results of the first two queries?*
- **Answer:** **1st query result: 1000 tuples; 2nd query result: 100000 tuples.**

Australian National University

SQL Queries – Group By Clause

- `GROUP BY` *attribute_list* groups tuples for each value combination in the attribute_list.
- Aggregate functions can be applied to aggregate a group of attribute values into a single value, e.g.,
  - `COUNT` returns the total number of argument values
  - `AVG` returns the average of argument values
  - `MIN` returns the minimum value of the arguments
  - `MAX` returns the maximum value of the arguments
  - `SUM` returns the sum of the argument values
- We can use `HAVING` *condition* to add the condition on the groups.

SQL Queries – Group By Clause

- List the total number of courses, the sum of the units of courses, the minimum unit in COURSE

COURSE		
No	Cname	Unit
COMP1130	Introduction to Advanced Computing I	6
COMP2400	Relational Databases	6
COMP3600	Algorithms	4

```
SELECT COUNT(*), SUM(unit), MIN(unit)
  FROM COURSE;
```

- The query result may look like:

COUNT	SUM	MIN
3	16	4

SQL Queries – Group By Clause

- List each course offered in Semester 2 2016 together with the number of students who have enrolled in the course

```
  SELECT e.CourseNo, COUNT(*) AS NumberOfStudents
    FROM Enrol e
   WHERE e.Semester = '2016 S2'
GROUP BY e.CourseNo ;
```

ENROL				
StudentID	CourseNo	Semester	Status	EnrolDate
458	COMP2400	2016 S2	active	25/02/2016
458	COMP1130	2016 S1	active	25/02/2016
456	COMP2400	2016 S2	active	25/02/2016
...	....	....	...	....

SQL Queries – Group By Clause

- List each course offered in Semester 2 2016 together with the number of students who have enrolled in the course

```
    SELECT e.CourseNo, COUNT(*) AS NumberOfStudents
      FROM Enrol e
     WHERE e.Semester = '2016 S2'
  GROUP BY e.CourseNo ;
```

- The query result may look like:

CourseNo	NumberOfStudents
COMP2400	120
COMP2600	100
COMP1130	150
...	...

SQL Queries – Having Clause

- List each course offered in Semester 2 2016 together with the number of students that is at least 120

```
  SELECT e.CourseNo, COUNT(*) AS NumberOfStudents
    FROM Enrol e
   WHERE e.Semester = '2016 S2'
GROUP BY e.CourseNo
  HAVING COUNT(*)>= 120 ;
```

- The query result may look like:

CourseNo	NumberOfStudents
COMP2400	120
COMP1130	150
...	...

SQL Queries – Order By Clause

- The ORDER BY clause allows us to sort the tuples in a query result.
  - ASC indicates ascending order (default).
  - DESC indicates descending order.
- We can sort the previous result by

```
  SELECT e.CourseNo, COUNT(*) AS NumberOfStudents
    FROM ENROL e
   WHERE e.Semester = '2016 S2'
GROUP BY e.CourseNo
ORDER BY NumberOfStudents DESC;
```

- This would return all tuples sorted by the number of enrolled students in descending order.

CourseNo	NumberOfStudents
COMP1130	150
COMP2400	120
COMP2600	100
...	...