SQL QUERIES

Aims: At the end of this group of four lectures, you should be able to use SQL to query your database.

Data retrieval in SQL
- Single table queries
- Multi-table queries
- Set operations
- Aggregate functions and grouping
- Complex queries

Reading: Elmasri & Navathe, Chapters 6 & 7

Slide 5-1

SELECT STATEMENT

- ***Not the same as*** the SELECT operation of the relational algebra
- Six clauses – only the first two mandatory
- SELECT <attribute list>
  FROM <table list>
  [WHERE <condition>]
  [GROUP BY <grouping attributes>]
  [HAVING <group condition>]
  [ORDER BY <attribute list>]
- *<attribute list>* is a list of attribute names whose values are to be retrieved by the query
- *<table list>* is a list of the relation names required to process the query
- *<condition>* is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query
- A query is evaluated by first taking the tables in FROM, applying the WHERE clause, then GROUP BY and HAVING, then SELECT clause, and finally ORDER BY

Slide 5-2

SIMPLE SQL QUERIES

- Basic SQL queries correspond to using the SELECT, PROJECT, and JOIN operations of the relational algebra

- **Query 1**: Show employee numbers only.
  SELECT EMP_NO
  FROM EMPLOYEE;

Emp_No	FName	LName	Address	Gender	Salary	DNO
111222333	John	Peters	33 Ilam Rd	M	45,000	7
123123123	Craig	Graham	56 Clyde Rd	M	80,000	2
444555666	Mary	Jones	100 Memorial Av	F	43,000	2
321321321	Ann	Stevens	354 Ferry Rd	F	68,000	7
777888999	Ian	Watkins	23 Hoani St	M	65,000	3

EMP_NO
111222333
123123123
444555666
321321321
777888999

RETRIEVALS: SIMPLE SEARCH CONDITIONS

Query 2: Show all information about employees working in department 2

```
SELECT *
FROM EMPLOYEE
WHERE DNO=2;
```

Emp_No	FName	LName	Address	Gender	Salary	DNO
111222333	John	Peters	33 Ilam Rd	M	45,000	7
123123123	Craig	Graham	56 Clyde Rd	M	80,000	2
444555666	Mary	Jones	100 Memorial Av	F	43,000	2
321321321	Ann	Stevens	354 Ferry Rd	F	68,000	7
777888999	Ian	Watkins	23 Hoani St	M	65,000	3

Emp_No	FName	LName	Address	Gender	Salary	DNO
123123123	Craig	Graham	56 Clyde Rd	M	80,000	2
444555666	Mary	Jones	100 Memorial Av	F	43,000	2

Slide 1-4

RETRIEVALS: COMPOUND CONDITIONS

Query 3: Show names of female employees who earn more than 60,000

```
SELECT FNAME, LNAME
FROM EMPLOYEE
WHERE Gender = 'F' AND
   Salary > 60000;
```

Emp_No	FName	LName	Address	Gender	Salary	DNO
111222333	John	Peters	33 Ilam Rd	M	45,000	7
123123123	Craig	Graham	56 Clyde Rd	M	80,000	2
444555666	Mary	Jones	100 Memorial Av	F	43,000	2
321321321	Ann	Stevens	354 Ferry Rd	F	68,000	7
777888999	Ian	Watkins	23 Hoani St	M	65,000	3

FName	LName
Ann	Stevens

Slide 1-5

SIMPLE SQL QUERIES

- **Query 4:** Comparison search condition
  List the names of all directors born in or after 1920.

- **Query 5:** Compound search condition (AND)
  List the titles and numbers of all movies that have won at least one Academy Award and have been made in or after 1988.

Slide 5-6

SIMPLE SQL QUERIES (CONT)

- **Query 6:** Compound search condition (OR)
  List the titles of all comedies or dramas.

- **Query 7:** NOT operator
  List the titles of all movies that have a critics rating.

Slide 5-7

ARITHMETIC OPERATIONS

- The standard arithmetic operators +, -, *, / can be applied to numeric values in an SQL query result
- **Query 8:** Calculated fields
  Produce a list of customer names, numbers and bonuses.

Slide 5-8

TABLES AS SETS IN SQL

- SQL does not automatically eliminate duplicate tuples in query results
- ALL (default) or DISTINCT in the `SELECT` clause

- **Query 9:**
  Produce a list of all stars that acted in movies no 5 or 6.

SIMPLE SQL QUERIES (CONT)

- **Query 10:** range search (BETWEEN)
  List the numbers and titles of all movies made between 2000 and 2005.

- **Query 11:** set membership (IN)
  List the numbers and titles of all movies whose type is comedy or drama.

Slide 5-10

PATTERN MATCHING

- *att* [NOT] **LIKE** *pattern*
- Patterns are formed by using two reserved characters:
  - % replaces an arbitrary number of characters (wild card)
  - _ replaces a single arbitrary character (position marker)
- Works with attributes of various data types
- **Query 12:** *Find all customers who live in Ilam.*

- **Query 13:** *Retrieve all directors who were born during the 1950s.*

- Escape character: 'Dark_Night%' escape '\'

Slide 5-11

REGULAR EXPRESSIONS

- Regular expressions added in SQL3
- REGEXP_LIKE (*att*, *pattern*, [*param*])
- Only string attributes can be used
- *param*
  - 'i' : case-insensitive
  - 'c' : case-sensitive
  - 'x' : ignore whitespace
- **Query 14:** Find all stars whose last name is Marais or Moranis.

Slide 5-12

NULLS IN SQL QUERIES

- Meanings of NULL
  - Unknown value
  - Unavailable or withheld value
  - Not applicable attribute
- SQL uses **IS NULL** or **IS NOT NULL** to compare NULLs because it considers each NULL value distinct from other NULL values, so equality comparison is not appropriate
- **Query 15:** NULL search condition
  List all directors who are still living.

Slide 5-13

THREE-VALUED LOGIC

Table 5.1 Logical Connectives in Three-Valued Logic

(a)	AND	TRUE	FALSE	UNKNOWN
	TRUE	TRUE	FALSE	UNKNOWN
	FALSE	FALSE	FALSE	FALSE
	UNKNOWN	UNKNOWN	FALSE	UNKNOWN
(b)	OR	TRUE	FALSE	UNKNOWN
	TRUE	TRUE	TRUE	TRUE
	FALSE	TRUE	FALSE	UNKNOWN
	UNKNOWN	TRUE	UNKNOWN	UNKNOWN
(c)	NOT			
	TRUE	FALSE		
	FALSE	TRUE		
	UNKNOWN	UNKNOWN		

ORDER BY

- Sort the tuples based on the values of some attribute(s)
- ASC (ascending, default) or DESC (descending)
- **Query 16:** Sorting results (ORDER BY)
  Produce a list of all movies, arranged in descending order of the number of Academy Awards won.

Slide 5-15

ORDER BY (CONT.)

- **Query 17:** Multiple column ordering
  List the numbers, names, addresses and join dates of all members. Sort the output by last name descending and by first name ascending.

Slide 5-16

AGGREGATE FUNCTIONS

- ***SUM, MIN, MAX, AVG, COUNT***
- NULL values ignored by aggregate functions
  - except COUNT(*)
- ALL and DISTINCT
- The SELECT clause may contain:
  - Column names
  - Aggregate functions
  - Constants
  - Expressions containing the above

Slide 5-17

AGGREGATE FUNCTIONS: EXAMPLES

- **Query 18:** *How many movies won more than four Academy Awards?*
- **Query 19:** *Find how many comedies there are and how many AA they won*

Slide 5-18

AGGREGATE FUNCTIONS: EXAMPLES

- **Query 20:** *How many types of movies are there in the database?*

Slide 5-19

GROUPING

- Often we want to apply the aggregate functions *to groups of tuples in a relation*
- Each group of tuples consists of the set of tuples that have *the same value* for the *grouping attribute(s)*
- The function is applied to each group independently
- **GROUP BY** clause specifies the grouping attributes
- **Query 21:** *Find the number of movies in each category and the total of AA won in each of them.*

Slide 5-20

GROUPING (CONT.)

- **Query 22:** *Show how many movies of each type each director has made.*
- **Query 23:** *For each director show how many different types of movies he/she has directed.*

Slide 5-21

HAVING CLAUSE

- Sometimes we want to retrieve the values of these functions for only those *groups that satisfy certain conditions*
- The HAVING clause is used for specifying a selection condition on groups (rather than on individual tuples)
- **Query 24:** *For each director, list the director's number and the total number of awards won by comedies he or she directed if that number is greater than 1.*

Slide 5-22

NESTING OF QUERIES

- A complete SELECT query, called a *nested query (a subquery)*, can be specified within the WHERE clause of another query (*the outer query)*
- **Query 25:** *List the numbers and names of all members who have rented more DVDs than average.*

Slide 5-23

NESTED QUERIES

- **Query 26:** *List the titles of all movies directed by Stanley Kubrick.*
- **Query 27:** *Show the names of all stars who acted in movie no 137.*

Slide 5-24

RULES FOR NESTED QUERIES

- ORDER BY not permitted in the nested query.
- The subquery SELECT list must consist of a single attribute or expression, except for queries using EXISTS.
- By default, attribute names in a subquery refer to the table names in the FROM clause of the subquery.
- A subquery can appear on both sides of a comparison, but some DBMSs allow nested queries only on the right-hand side.
- A subquery may not be used as an operand in an expression.
- In general, there can be several levels of nesting.

Slide 5-25

CORRELATED NESTED QUERIES

- If a condition in the WHERE-clause of a *nested query* references an attribute of a relation declared in the *outer query*, the two queries are said to be *correlated*
- The result of a correlated nested query is *different for each tuple (or combination of tuples) of the relation(s) the outer query*

Slide 5-26

CORRELATED NESTED QUERIES (CONT)

- **Query 28:** *For each director, his/her number and the title of his/her movie that got most Academy Awards.*

Slide 5-27

EXISTS

- EXISTS is used to check whether the result of a correlated nested query is empty (contains no tuples) or not
- **Query 29:** *Find the numbers and names of all directors who have directed at least one comedy.*

Slide 5-28

EXISTS (CONT.)

- **Query 30:** *Find the names of directors who made no dramas.*

Slide 5-29

MULTITABLE QUERIES

- A query containing a nested query and using the = or IN conditions can ***always*** be expressed as a single query.
- In that case, the FROM clause of the query may contain more than one table
- If there is more than one table in FROM, join conditions are necessary
- **Query 26 revisited:**
  List the titles of all movies directed by Stanley Kubrick.

Slide 5-30

UNSPECIFIED WHERE CLAUSE

- A *missing WHERE-clause* indicates no condition
- All tuples of the relations in the FROM clause are selected
- This is equivalent to the condition WHERE TRUE
- If more than one relation is specified FROM *and* there is no join condition, then the *CARTESIAN PRODUCT* of tuples is selected
- It is extremely important not to overlook specifying any selection and join conditions in the WHERE clause; otherwise, incorrect and very large relations may result
- **Query 31:** *Select all combinations of movie titles and DVD codes.*

Slide 5-31

MULTIPLE TABLE JOIN

- The join condition may be specified in WHERE
- SQL2 also allows joins to be specified in FROM

  FROM table1 join-type JOIN table2 ON att1=att2
- **Query 26 revisited:**
  List the titles of all movies directed by Stanley Kubrick.

Slide 5-32

MULTIPLE TABLE JOIN

- **Query 32:** *List the names and addresses of all customers currently renting Mel Gibson's movies.*

Slide 5-33

ALIASES

- Some queries need to refer to the same relation twice
- In this case, an *alias* (i.e. alternative name) is given to the relation
- Can also use the AS keyword to specify aliases
- Aliasing can also be used in any SQL query for convenience
- **Query 33:** *Find the list of any pairs of stars who have the same last name.*

Slide 5-34

SPECIFYING JOINS IN SQL2

- FROM table1 join-type JOIN table2 ON att1=att2
- Join-type: *join, natural join, left outer join, right outer join, full outer join, cross join*
- Outer join in Oracle:
  - Additionally, the (+) operator can be used
  - Use (+) after the join attribute which might need NULLs to be added
- **Query 34:** *List names and numbers of all members. For those who are currently renting DVDs, list the number of DVDs too.*

Slide 5-35

JOINED RELATIONS IN SQL2

- Can specify a "joined relation" in the FROM clause
- Looks like any other relation but is the result of a join
- **Query 26 revisited:** *List the titles of all movies directed by Stanley Kubrick.*

```
SELECT  TITLE
  FROM  MOVIE, DIRECTOR
  WHERE  DIRECTOR=DNUMBER and LNAME='Kubrick'
        and FNAME='Stanley'
```

can be written as:

```
SELECT  TITLE
FROM    (select * from MOVIE join DIRECTOR on
              DIRECTOR=DNUMBER) MOVDIR
WHERE movdir.lname='Kubrick' and movdir.fname='Stanley'
```

Slide 5-36

SET OPERATIONS

- **UNION**
- **EXCEPT (MINUS in Oracle)**
- **INTERSECT**

- The resulting relations of these set operations are sets of tuples; *duplicate tuples are eliminated from the result*
- Corresponding multiset operations: `UNION ALL`, `EXCEPT ALL`, `INTERSECT ALL`
- Arguments must be *union compatible relations*

Slide 5-37

SET OPERATIONS (CONT)

- **Query 35:** *List the DVD codes and movie numbers for all DVDs on which the movie is a comedy or that are rented by Mark Haley.*

Slide 5-38

SET OPERATIONS (CONT)

- **Query 36:** *Find the numbers of directors who have directed both comedies and dramas.*

Slide 5-39

SET OPERATIONS (CONT)

- **Query 37:** *Find the numbers of directors who have directed comedies, but not dramas.*

Slide 5-40

ANY/ALL PREDICATE

- Used with a comparison operator
- In some DBMSs, SOME used instead of ANY
- !=ALL is equivalent to NOT IN
- =ANY is equivalent to IN
- **Query 38:** *List numbers and titles of all movies that were nominated for more awards than any movie directed by Woody Allen.*

Slide 5-41

DIVISION IN SQL

- **Query 39:** *Find the names of directors who have directed at least one movie of each type.*

Slide 5-42

DUAL TABLE

- Available to all users
- Has one column (dummy, varchar(1)) and one row with a value X
- Useful for getting a constant expression

```
Select sysdate from dual;
```

Slide 5-43