CSC375/BIF375
Database Management Systems

Chapter 6
The Database Language SQL
Part 1 - Simple Queries



Why SQL?

- **SQL** (pronounced "ess-que-el") stands for Structured Query Language
- **SQL** is a very-high-level language.
  - Say “what to do” rather than “how to do it.”
  - Avoid a lot of data-manipulation details needed in procedural languages like C++ or Java.
- DBMS figures out “best” way to execute query: query optimization

Basic keywords

Mandatory

SELECT

To select all or some attribute(s)/ columns from the table:

Mandatory

FROM

To select which table(s) are we working on

Optional

WHERE

Condition(s) to select rows from the table:

Examples

- Consider the table called:

students (name, lastname, age, ID)

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Example 1: projection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name
FROM students
```

Result

name
rim
sara
milad
tarek

```
SELECT age
FROM students
```

Result

age
19
18
21
18

```
SELECT name, age
FROM students
```

Result

name	age
rim	19
sara	18
milad	21
tarek	18

Example 1: projection (Cont.)

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Task

What if we want to project *all* the columns of the table?

option 1

```
SELECT name, lastname, age, ID
FROM students
```

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

option 2

`*` means select all columns of the table

```
SELECT *
FROM students
```

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

Example 2: Extended projection

- Projecting attributes into different names
- Adding simple arithmetic while projecting
- Keyword: **`AS`**
- Results are temporary, original table is unchanged.

Example 2: Extended projection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT lastname AS familyName,
       age-18 AS legalSince
FROM students
```

Result

familyName	legalSince
ismail	1
kamel	0
ghantous	3
dib	0

Example 2: Extended projection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
--constant values in a query
SELECT lastname AS familyName,
       'student'
FROM students
```

Result

familyName	student
ismail	student
kamel	student
ghantous	student
dib	student


QUIZ
TIME!

Consider this table `t`

a	b	c
1	2	7
3	4	12
2	3	20

What's the output of the following queries?

```
SELECT a
FROM t
```

```
SELECT *
FROM t
```

```
SELECT a+b AS n
FROM T
```

Example 3: Selection

- Keyword: **`WHERE`**
- Add any condition(s) you wish to be applied on a **row by row** basis
- You can use **`AND`**, **`OR`** to combine your conditions
- **`NOT`** keyword can be used for complement
- Only rows that satisfy the condition(s) are returned.

Example 3: Selection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name, lastname
FROM students
WHERE ID=123
```

Result

name	lastname
rim	ismail

```
SELECT lastname
FROM students
WHERE name= 'sara'
```

Result

lastname
kamel

Example 3: Selection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name, lastname
FROM students
WHERE age > 18
```

Result

name	lastname
rim	ismail
milad	ghantous

Example 3: Selection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name, lastname
FROM students
WHERE NOT (age > 18)
```

Result

name	lastname
sara	kamel
tarek	dib

Example 3: Selection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
WHERE age > 18
      AND
      ID > 1000
```

Result

name	lastname	age	ID
milad	ghantous	21	1345

Example 3: Selection

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
WHERE age > 18
      OR
      ID > 1000
```

Result

name	lastname	age	ID
rim	ismail	19	123
milad	ghantous	21	1345
tarek	dib	18	12345

Example 3: Selection

Relation R

x	y
1	A
2	B
3	NULL

```
SELECT *
FROM R
WHERE y is NULL
```

x	y
3	NULL

```
SELECT *
FROM R
WHERE y is NOT NULL
```

x	y
1	A
2	B

Comparison operators

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>, !=	Not equal to
IS NOT NULL	NOT NULL value test
IS NULL	NULL value test

Example 3: Selection – IN operator

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
WHERE age in (19, 21)
```

Result

name	lastname	age	ID
rim	ismail	19	123
milad	ghantous	21	1345

```
SELECT *
FROM students
WHERE age not in (19, 21)
```

name	lastname	age	ID
sara	kamel	18	134
tarek	dib	18	12345

Order of execution

- What do you think is the order of execution of SELECT FROM WHERE?

- Which one happens first? second? last?

Answer

- **FROM** is executed first to select the table(s) we are working with.
- **WHERE** is executed second to return only the rows satisfying the conditions
- **SELECT** is executes last to project the columns needed.


QUIZ
TIME!

Consider this table **t**

a	b	c
1	2	7
3	4	12
2	3	20

What's the output of the following queries?

```
SELECT a
FROM t
WHERE a>1
```

```
SELECT a
FROM t
WHERE b>=3
```

```
SELECT *
FROM T
WHERE a>1 and b>3
```

Pattern Matching

- Keywords: **`LIKE`**, **`%`**, **`_`** (underscore)
- Used to match specific words or parts of words in the database.
- For example, if we want to search for all strings that start with “M” or that end with “S”, or that contain “a” in the string.
- Also, we can search for strings that are 3 characters long for example.
- Combine both of the above.

Pattern matching- part 1

- Use `%` to denote any combination of characters (could also match zero characters)

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
WHERE lastname LIKE 'i%'
```

This means all last names that start with letter i and anything after

Result

name	lastname	age	ID
rim	ismail	19	123

Pattern matching- part 1

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name, lastname
FROM students
WHERE lastname LIKE '%l'
```

This means all last names that end with letter l

Result

name	lastname
rim	ismail
sara	kamel

Pattern matching- part 1

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name, lastname
FROM students
WHERE lastname LIKE '%m%'
```

This means all last names that **contain** the letter m

Result

name	lastname
rim	ismail
sara	kamel

Pattern matching- part 2

- Use **_ (underscore)** to denote ONE character

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name, lastname
FROM students
WHERE lastname LIKE '_ _ _ _ _l'
```

all last names that have exactly 5 characters followed by l

Result

name	lastname
rim	ismail

only ismail succeeds because kamel has 4 characters before letter l not 5


QUIZ
TIME!

Quiz

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT name
FROM students
WHERE name LIKE '%a%'
```

Output?

Write a query that outputs names containing the letter **'a'** followed by 2 characters only.

```
SELECT name
FROM students
WHERE name LIKE '%a_ _'
```

`Order by` clause

- This is an optional clause to be used with the query
- It sorts the rows depending on what we provide to it either **alphabetically** or **numerically**.
- The **default** is ascending order but we can specify descending order using the keyword **`DESC`**
- You can sort using more than 1 attribute

Ordering-part 1

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
ORDER BY name
```

Sort the rows alphabetically using the `name` attribute in ascending order

Result

name	lastname	age	ID
milad	ghantous	21	1345
rim	ismail	19	123
sara	kamel	18	134
tarek	dib	18	12345

Ordering-part 2

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
ORDER BY name DESC
```

Sort the rows alphabetically using the `name` attribute in descending order

Result

name	lastname	age	ID
tarek	dib	18	12345
sara	kamel	18	134
rim	ismail	19	123
milad	ghantous	21	1345

Ordering-part 2

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
ORDER BY age, ID
```

Start with sorting according to age, if there's a tie, use ID to sort
All in ascending order

Result

name	lastname	age	ID
sara	kamel	18	134
tarek	dib	18	12345
rim	ismail	19	123
milad	ghantous	21	1345


QUIZ
TIME!

Quiz

name	lastname	age	ID
rim	ismail	19	123
sara	kamel	18	134
milad	ghantous	21	1345
tarek	dib	18	12345

```
SELECT *
FROM students
ORDER BY age, lastname
```

Result

End of part 1