

Tishk International University  
Department of Information Technology  
Database Systems 1  
Week 5  
Fall 2023-24  
October 29, 2023



# Queries

Wisam Abdulaziz Qadir  
Wisam.abdulaziz@tiu.edu.iq

# Objectives



- What is Query
- What is SQL
- Basic SQL Query Clauses
- Basic SQL Operations
- Basic SQL Functions

# Query



- It is a request for actions on data in database.
- It can be used to:
  - Create and delete databases
  - create and delete tables
  - insert records to a database
  - retrieve data from a database
  - update records in a database
  - delete records from a database
  - etc.

# Structured Query Language (SQL)



- It is a language for accessing and manipulating databases.
- It is used to communicate with databases.
- MS Access uses SQL.

# Basic Query Structure



- A typical SQL query has the form:

```
SELECT   $A_1, A_2, \dots, A_n$   
FROM     $r_1, r_2, \dots, r_m$   
WHERE    $P$ 
```

- $A_i$  represents an attribute (row)
  - $R_i$  represents a relation (table)
  - $P$  is a predicate (condition)
- 
- The result of an SQL query is a **relation**.

# The *select* Clause



- The SQL **SELECT** statement returns a set of records from one or more tables.
- Example: find first name of all instructors:

```
select first_name  
from instructor;
```

- NOTE: SQL names are case insensitive (i.e., you may use upper or lower case letters.)
  - E.g. *First\_name*  $\equiv$  *FIRST\_NAME*

# The select Clause (cont.)



- Example: find first name of all students.

```
select F_name  
from Student;
```

<u>SID</u>	F_name	L_name	Mark	Age
1	Dara	Azad	90	20
2	Zara	Nawzad	95	22
3	Dara	Kawa	85	19

**Student**



F_name
Dara
Zara
Dara

**Query result**

# The `select` Clause (cont.)



- SQL allows duplicates in relations as well as in query results.
- To force the elimination of duplicates, insert the keyword **distinct** after **select**.



# The select Clause (cont.)



- Example: find first name of all students without any duplicates.

```
select distinct F_name  
from Student;
```

<u>SID</u>	F_name	L_name	Mark	Age
1	Dara	Azad	90	20
2	Zara	Nawzad	95	22
3	Dara	Kawa	85	19

**Student**



F_name
Dara
Zara

**Query result**

# The select Clause (cont.)



- An asterisk in the select clause denotes “all attributes (fields)”.

```
select *  
from Employee;
```

<u>emp_id</u>	first_name	salary
123	Dara	800
444	Zara	1000
555	Nasrin	850

Employee



<u>emp_id</u>	first_name	salary
123	Dara	800
444	Zara	1000
555	Nasrin	850

Query result

# The select Clause (cont.)



- Arithmetic operations can be written within select clause to perform calculations.

```
select first_name, salary/2  
from employee;
```

<u>emp_id</u>	first_name	salary
123	Dara	800
444	Zara	1000
555	Nasrin	850

**Employee**



first_name	salary
Dara	400
Zara	500
Nasrin	425

**Query result**

# The Where Clause



- The SQL **where** clause specifies condition(s) that the result must satisfy.
- E.g.: Find the first names and marks of all students whose marks are greater than 85.

```
select F_name, Mark
from Student
where Mark > 85;
```

<u>SID</u>	F_name	L_name	Mark	Age
1	Dara	Azad	90	20
2	Zara	Nawzad	95	22
3	Nasrin	Kawa	85	19

Student



F_name	Mark
Azad	90
Nawzad	95

Query result

# The Where Clause (cont.)



- Comparison results can be combined using the logical connectives **and**, **or**, and **not**.
- E.g.: Find the first names and marks of all students whose marks are greater than 85 and Ages are greater than 20.

```
select F_name, Mark  
from Student  
where Mark > 85 and Age > 20;
```

<u>SID</u>	F_name	L_name	Mark	Age
1	Dara	Azad	90	20
2	Zara	Nawzad	95	22
3	Nasrin	Kawa	85	19

**Student**



F_name	Mark
Zara	95

**Query result**

# The Where Clause (cont.)



- E.g.: Find mark and age of the student whose first name is Dara and last name is Kawa.

```
select Mark, Age  
from Student  
where F_name = 'Dara' and L_name = 'Kawa';
```

<u>SID</u>	F_name	L_name	Mark	Age
1	Dara	Azad	90	20
2	Zara	Nawzad	95	22
3	Dara	Kawa	85	19

**Student**



Mark	Age
85	19

**Query result**

# The Where Clause (cont.)



- SQL includes **between** comparison operator.
- E.g.: Find first name and age of all students whose ages are between 18 and 21.

```
select F_name, Age
from Student
where Age between 18 and 21;
```

<u>SID</u>	F_name	L_name	Mark	Age
1	Dara	Azad	90	20
2	Zara	Nawzad	95	22
3	Nasrin	Kawa	85	19

Student



F_name	Age
Dara	20
Nasrin	19

Query result

# The Where Clause (cont.)



- The predicate **is null** is used to check for null values.
- E.g.: Find first name of all students whose marks are not written.

```
select F_name  
from Student  
where Mark is null;
```

<u>SID</u>	F_name	L_name	Mark	Age
1	Dara	Azad	90	20
2	Zara	Nawzad		22
3	Nasrin	Kawa	85	19

**Student**



F_name
Zara

**Query result**



# Aliases (Renaming)



- The SQL allows renaming of relations and attributes using the **as** clause:

Syntax: old-name **as** new-name

- E.g.
  - **select** *name, salary/12 as monthly\_salary*  
**from** *Employee;*
  - **select** name, salary/12  
**from** *Employee as E;*

# Ordering Operation



- It is used to order the result of the query.

```
select *  
from Student  
order by F_name;
```

<u>SID</u>	F_name	L_name	Mark
1	Dara	Azad	90
2	Zara	Nawzad	95
3	Nasrin	Kawa	85

Student



<u>SID</u>	F_name	L_name	Mark
1	Dara	Azad	90
3	Nasrin	Kawa	85
2	Zara	Nawzad	95

Query result

# Ordering Operation (cont.)



- **asc** for ascending and **desc** for descending order can be specified.
- **asc** order is the default.
  - Example: **order by** *name* **desc**
- Can sort on multiple attributes
- Example: **order by** *dept\_name, name*

# Aggregate Functions



- These functions operate on the multiset of values of a field in a relation and return a value.

Syntax: **select Aggregate\_Functions** (field name)

**avg** : average value

**min** : minimum value

**max** : maximum value

**sum** : sum of values

**count** : number of values

# Aggregate Functions (cont.)



- Find the average salary of all instructors.

```
select avg (salary)  
from instructor;
```

- Find the total number of students

```
select count (F_name)  
from students;
```

- Find the total salary of instructors

```
select sum (salary)  
from instructor;
```

# Aggregate Functions (cont.)



- Find the minimum salary among all the instructors.

```
select min (salary)  
from instructor;
```

- Find the maximum salary among all the instructors.

```
select max (salary)  
from instructor;
```

# Group By



- Find the average salary of instructors in each department.

```
select dept_name, avg (salary)  
from instructor  
group by dept_name;
```

<u>ID</u>	Instructor_name	Dept_name	Salary
1	Dara	IT	800
2	Zara	Civil	1000
3	Nasrin	Civil	850
4	Ali	Biology	500
5	Kawa	Biology	700

**Instructor**



Dept_name	salary
IT	800
Civil	925
Biology	600

**Query result**



Thank You