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





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





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





- 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 $A_1, A_2, ..., A_n$ **FROM** $r_1, r_2, ..., r_m$ **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 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 = FIRST_NAME*

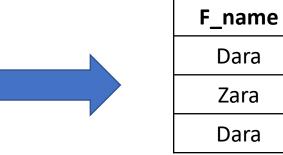




• Example: find first name of all students.

select F_name from Student;

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



Student

Query result





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





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

select distinct F_name from Student;

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



Query result





• An asterisk in the select clause denotes "all attributes (fields)".

select *
from Employee;

emp_id	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	first_name	salary
123	Dara	800	Dara	400
444	Zara	1000	Zara	500
555	Nasrin	850	Nasrin	425

Employee

Query result





- 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;

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



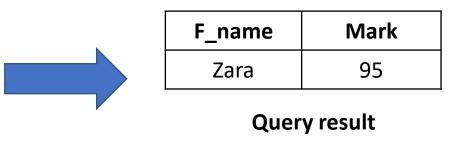
Query result



- 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;

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

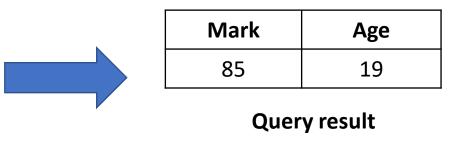




• E.g.: Find mark and age of the student whose first name is Dara and last name is Kawa.

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

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

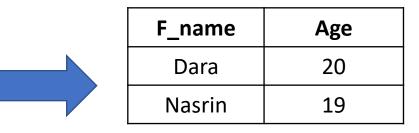




- SQL includes **between** comparison operator.
- E.g.: Find first name and age of all students whose ages are between 18 and 21.

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

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



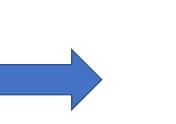
Query result



- 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;

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



F_	name
	Zara

Query result





• 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;

SID	F_name	L_name	Mark		SID	F_name	L_name	Mark
1	Dara	Azad	90		1	Dara	Azad	90
2	Zara	Nawzad	95		3	Nasrin	Kawa	85
3	Nasrin	Kawa	85		2	Zara	Nawzad	95
Student			I		Qı	ery 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*





• 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;





• Find the average salary of instructors in each department.

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

ID	Instructor_name	Dept_name	Salary
1	Dara	IT	800
- -			
Ζ	Zara	Civil	1000
3	Nasrin	Civil	850
4	Ali	Biology	500
5	Kawa	Biology	700

Instructor



