Lecture (9)
"Introduction to Data Base“
SQL Database

Presented by : Dr. Nehal El Azaly
Dr. Dina Abdel hafez

Introduction to SQL for Beginners

One to one Example using SQL

queries.sql + 3zvgp366v NEW MYSQL RUN

```sql
-- create
CREATE TABLE EMPLOYEE (
  EMPLOYEE_id INTEGER PRIMARY KEY,
  name varchar(255) NOT NULL,
  ADDRESS varchar(500) NOT NULL,
  CITY VARCHAR(255)
);
CREATE TABLE PAYROLL (
  PAYROLL_ID INT PRIMARY KEY,
  SALARY INT NOT NULL,
  POSITION VARCHAR(255) NOT NULL,
  -- PersonID int FOREIGN KEY REFERENCES Persons(PersonID),
  EMPOLYEE_ID INT UNIQUE REFERENCES EMPLOYEE(EMPLOYEE_id)
  ON UPDATE CASCADE
  ON DELETE CASCADE
);


insert into EMPLOYEE
(EMPLOYEE_id, name, ADDRESS)
values(1, "mohamed", "alexandria"),
(2, "ahmed", "alexandria");

select * from EMPLOYEE;


insert into payroll
(PAYROLL_ID, SALARY, POSITION, EMPOLYEE_ID)
values (1, 1000, "developer", 1), (2, 3000, "anthing", 2);

select * from payroll;
```

Set the relation

Search on the PK record

STDIN

Input for the program (Optional)

Output:

```
EMPLOYEE_id     name    ADDRESS CITY
1       mohamed alexandria      NULL
2       ahmed   alexandria      NULL
PAYROLL_ID      SALARY  POSITION        EMPOLYEE_ID
1       1000    developer       1
2       3000    anthing 2
```

One to many Example using SQL

queries.sql + 3zvgp366v NEW MYSQL RUN

```sql
CREATE TABLE CUSTOMERS
(
 CUSTOMER_ID INT AUTO_INCREMENT PRIMARY KEY,
 CUSTOMER_NAME VARCHAR(255) NOT NULL,
 PHONE_NUMBER VARCHAR(255) NOT NULL,
 ADDRESS VARCHAR(500)
);

CREATE TABLE ORDERS
(
  ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
  ORDER_DATE DATE,
  COST INT NOT NULL,
  CUSTOMER_ID INT NOT NULL,
  foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
);
```

STDIN

Input for the program (Optional)

Output:

Program did not output anything!

queries.sql + 3zvgrvw8p NEW MYSQL RUN

```sql
CREATE TABLE CUSTOMERS
(
 CUSTOMER_ID INT AUTO_INCREMENT PRIMARY KEY,
 CUSTOMER_NAME VARCHAR(255) NOT NULL,
 PHONE_NUMBER VARCHAR(255) NOT NULL,
 ADDRESS VARCHAR(500)
);

CREATE TABLE ORDERS
(
  ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
  ORDER_DATE DATE,
  COST INT NOT NULL,
  CUSTOMER_ID INT NOT NULL,
  foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
);

INSERT INTO CUSTOMERS
(CUSTOMER_NAME, PHONE_NUMBER, ADDRESS)
VALUES ("MOHAMED", "+2011111111", "ALEXANDRIA"),
       ("AHMED", "+201233333333", "ALEXANDRIA");

INSERT INTO ORDERS
(ORDER_DATE, COST, CUSTOMER_ID)
VALUES ("2023-12-4", 4000, 1), ("2023-12-5", 5000, 2);


SELECT * FROM CUSTOMERS;
SELECT * FROM ORDERS;
```

STDIN

Input for the program (Optional)

Output:

CUSTOMER_ID	CUSTOMER_NAME	PHONE_NUMBER	ADDRESS
1	MOHAMED	+2011111111	ALEXANDRIA
2	AHMED	+201233333333	ALEXANDRIA

ORDER_ID	ORDER_DATE	COST	CUSTOMER_ID
1	2023-12-04	4000	1
2	2023-12-05	5000	2

Ascending sorting (Order By)

queries.sql + 3zw47atvv NEW MYSQL RUN

```sql

--
CREATE TABLE EMPLOYEE (
  EMP_ID INT AUTO_INCREMENT PRIMARY KEY,
  NAME VARCHAR(255) NOT NULL,
  ADDRESS VARCHAR(500) NOT NULL,
  CITY VARCHAR(255)
);

-- insert
INSERT INTO EMPLOYEE (NAME, ADDRESS)
VALUES ('Clark', 'ALEX'),
       ('Dave', 'ALEX'),
       ('Ava', 'ALEX');

-- fetch
SELECT * FROM EMPLOYEE
ORDER BY name ASC;

```

STDIN

Input for the program (Optional)

Output:

EMP_ID	NAME	ADDRESS	CITY
3	Ava	ALEX	NULL
1	Clark	ALEX	NULL
2	Dave	ALEX	NULL

Descending sorting (Order By)

queries.sql + 3zw47atvv NEW MYSQL RUN

```
1
2  --
3  CREATE TABLE EMPLOYEE (
4    EMP_ID INT AUTO_INCREMENT PRIMARY KEY,
5    NAME VARCHAR(255) NOT NULL,
6    ADDRESS VARCHAR(500) NOT NULL,
7    CITY VARCHAR(255)
8  );
9
10 -- insert
11 INSERT INTO EMPLOYEE (NAME, ADDRESS)
12 VALUES ('Clark', 'ALEX'),
13        ('Dave', 'ALEX'),
14        ('Ava', 'ALEX');
15
16 -- fetch
17 SELECT * FROM EMPLOYEE
18 ORDER BY name desc;
19
```

STDIN

Input for the program (Optional)

Output:

EMP_ID	NAME	ADDRESS	CITY
2	Dave	ALEX	NULL
1	Clark	ALEX	NULL
3	Ava	ALEX	NULL

Where clause condition

queries.sql + 3zw47atvv NEW MYSQL RUN

```sql
--
CREATE TABLE EMPLOYEE (
  EMP_ID INT AUTO_INCREMENT PRIMARY KEY,
  NAME VARCHAR(255) NOT NULL,
  ADDRESS VARCHAR(500) NOT NULL,
  CITY VARCHAR(255)
);

-- insert
INSERT INTO EMPLOYEE (NAME, ADDRESS)
VALUES ('Clark', 'ALEX'),
       ('Ibrahim', 'ALEX'),
       ('Sherif', 'Cairo'),
       ('Mohamed', 'Cairo'),
       ('ALI', 'Cairo'),
       ('Ahmed', 'Cairo'),
       ('Ava', 'aswan'),
       ('Salma', 'aswan');

-- fetch
SELECT * FROM EMPLOYEE
WHERE ADDRESS = 'Cairo';
```

STDIN ctrl + enter

Input for the program (Optional)

Output:

EMP_ID	NAME	ADDRESS	CITY
3	Sherif	Cairo	NULL
4	Mohamed	Cairo	NULL
5	ALI	Cairo	NULL
6	Ahmed	Cairo	NULL

Where clause condition with comparison operators

queries.sql + 3zw47atvv NEW MYSQL RUN

```sql

--
CREATE TABLE EMPLOYEE (
  EMP_ID INT AUTO_INCREMENT PRIMARY KEY,
  NAME VARCHAR(255) NOT NULL,
  ADDRESS VARCHAR(500) NOT NULL,
  Salary INT NOT NULL,
  CITY VARCHAR(255)
);

-- insert
INSERT INTO EMPLOYEE (NAME, ADDRESS, Salary)
VALUES ('Clark', 'ALEX', 5000 ),
       ('Ibrahim', 'ALEX', 4000),
       ('Sherif', 'Cairo', 10000),
       ('Mohamed', 'Cairo', 7000),
       ('ALI', 'Cairo', 3000),
       ('Ahmed', 'Cairo', 13000),
       ('Ava', 'aswan', 12000),
       ('Salma', 'aswan', 8000);

-- fetch
SELECT * FROM EMPLOYEE
WHERE ADDRESS = 'Cairo' AND Salary >= 5000;

```

STDIN

Input for the program (Optional)

Output:

EMP_ID	NAME	ADDRESS	Salary	CITY
3	Sherif	Cairo	10000	NULL
4	Mohamed	Cairo	7000	NULL
6	Ahmed	Cairo	13000	NULL

Where clause with logic operators

queries.sql + 3zw47atvv NEW MYSQL RUN

```sql
--
CREATE TABLE EMPLOYEE (
  EMP_ID INT AUTO_INCREMENT PRIMARY KEY,
  NAME VARCHAR(255) NOT NULL,
  ADDRESS VARCHAR(500) NOT NULL,
  Salary INT NOT NULL,
  CITY VARCHAR(255)
);

-- insert
INSERT INTO EMPLOYEE (NAME, ADDRESS, Salary)
VALUES ('Clark', 'ALEX', 5000 ),
       ('Ibrahim', 'ALEX', 4000),
       ('Sherif', 'Cairo', 10000),
       ('Mohamed', 'Cairo', 7000),
       ('ALI', 'Cairo', 3000),
       ('Ahmed', 'Cairo', 13000),
       ('Ava', 'aswan', 12000),
       ('Salma', 'aswan', 8000);

-- fetch
SELECT * FROM EMPLOYEE
WHERE (ADDRESS = 'Cairo' Or ADDRESS = 'Alex') and Salary >= 5000;
```

STDIN

Input for the program (Optional)

Output:

EMP_ID	NAME	ADDRESS	Salary	CITY
1	Clark	ALEX	5000	NULL
3	Sherif	Cairo	10000	NULL
4	Mohamed	Cairo	7000	NULL
6	Ahmed	Cairo	13000	NULL

Where clause with logic operators

queries.sql + 3zw47atvv NEW MYSQL RUN

```sql

--
CREATE TABLE EMPLOYEE (
  EMP_ID INT AUTO_INCREMENT PRIMARY KEY,
  NAME VARCHAR(255) NOT NULL,
  ADDRESS VARCHAR(500) NOT NULL,
  Salary INT NOT NULL,
  CITY VARCHAR(255)
);

-- insert
INSERT INTO EMPLOYEE (NAME, ADDRESS, Salary)
VALUES ('Clark', 'ALEX', 5000 ),
       ('Ibrahim', 'ALEX', 4000),
       ('Sherif', 'Cairo', 10000),
       ('Mohamed', 'Cairo', 7000),
       ('ALI', 'Cairo', 3000),
       ('Ahmed', 'Cairo', 13000),
       ('Ava', 'aswan', 12000),
       ('Salma', 'aswan', 8000);

-- fetch
SELECT * FROM EMPLOYEE
WHERE NOT ADDRESS = 'ALEX' and Salary >= 5000;

```

STDIN

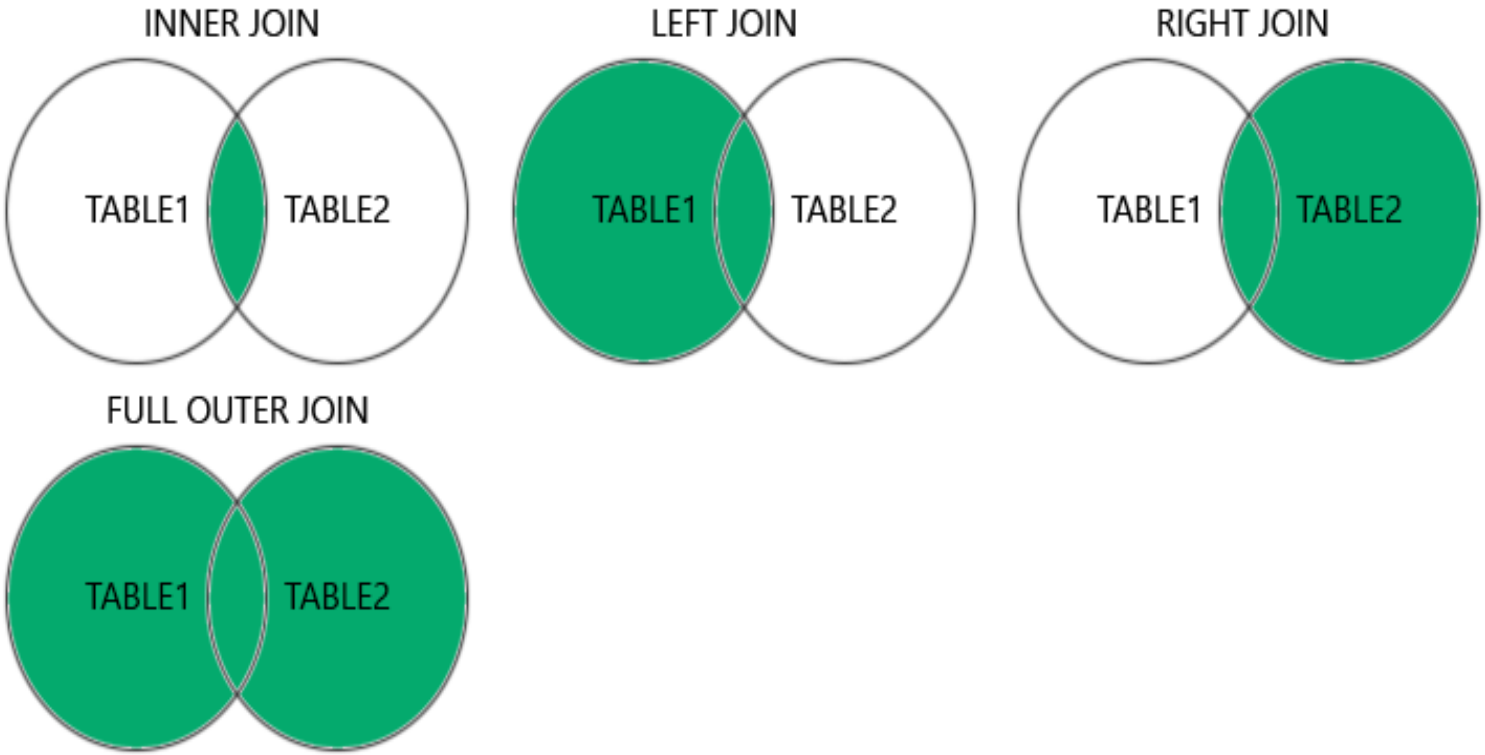
Input for the program (Optional)

Output:

EMP_ID	NAME	ADDRESS	Salary	CITY
3	Sherif	Cairo	10000	NULL
4	Mohamed	Cairo	7000	NULL
6	Ahmed	Cairo	13000	NULL
7	Ava	aswan	12000	NULL
8	Salma	aswan	8000	NULL

Different Types of SQL joins

- `(INNER) JOIN` : Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN` : Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN` : Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN` : Returns all records when there is a match in either left or right table



Inner join for one to many Relation ship

queries.sql + 3zw47atvv NEW MYSQL RUN

```sql
--
CREATE TABLE CUSTOMERS (
  CUSTOMER_ID INT AUTO_INCREMENT PRIMARY KEY,
  NAME VARCHAR(255) NOT NULL,
  ADDRESS VARCHAR(500) NOT NULL
);

CREATE TABLE ORDERS
(
  ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
  ORDER_DATE DATE NOT NULL,
  COST INT NOT NULL,
  CUSTOMER_ID INT NOT NULL,
  foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
);
-- insert
INSERT INTO CUSTOMERS (NAME, ADDRESS)
VALUES ('Clark', 'ALEX'),
       ('Ibrahim', 'ALEX'),
       ('Sherif', 'Cairo'),
       ('Mohamed', 'Cairo'),
       ('ALI', 'Cairo'),
       ('Ahmed', 'Cairo'),
       ('Ava', 'aswan'),
       ('Salma', 'aswan');

INSERT INTO ORDERS (ORDER_DATE, COST, CUSTOMER_ID)
VALUES
  ('2023-12-11', 5000, 1),
  ('2023-12-11', 6000, 1),
  ('2023-12-11', 5000, 3),
  ('2023-12-11', 5000, 2),
  ('2023-12-11', 5000, 4);
```

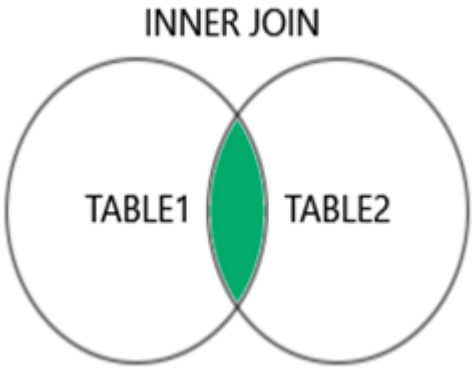
STDIN

Input for the program (Optional)

ctrl + enter

Output:

ORDER_ID	ORDER_DATE	COST	CUSTOMER_ID	CUSTOMER_ID	
1	2023-12-11	5000	1	1	Clark ALEX
2	2023-12-11	6000	1	1	Clark ALEX
3	2023-12-11	5000	3	3	Sherif Cairo
4	2023-12-11	5000	2	2	Ibrahim ALEX
5	2023-12-11	5000	4	4	Mohamed Cairo



Inner join

queries.sql + 3zw47atvv NEW MYSQL RUN

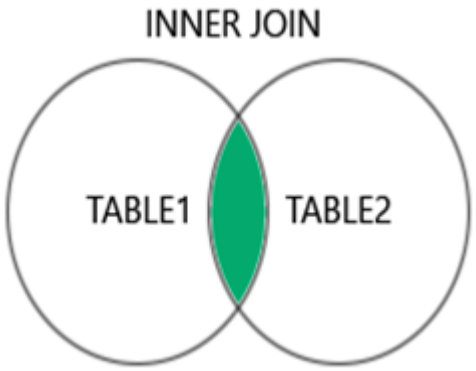
```
  ADDRESS VARCHAR(500) NOT NULL
);

CREATE TABLE ORDERS
(
  ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
  ORDER_DATE DATE NOT NULL,
  COST INT NOT NULL,
  CUSTOMER_ID INT NOT NULL,
  foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
);
-- insert
INSERT INTO CUSTOMERS (NAME, ADDRESS)
VALUES ('Clark', 'ALEX'),
       ('Ibrahim', 'ALEX'),
       ('Sherif', 'Cairo'),
       ('Mohamed', 'Cairo'),
       ('ALI', 'Cairo'),
       ('Ahmed', 'Cairo'),
       ('Ava', 'aswan'),
       ('Salma', 'aswan');

INSERT INTO ORDERS (ORDER_DATE, COST, CUSTOMER_ID)
VALUES
  ('2023-12-11', 5000, 1),
  ('2023-12-11', 6000, 1),
  ('2023-12-11', 5000, 3),
  ('2023-12-11', 5000, 2),
  ('2023-12-11', 5000, 4);

SELECT * FROM ORDERS
INNER JOIN Customers ON ORDERS.Customer_ID = Customers.Customer_ID;
```

STDIN ctrl + enter

Input for the program (Optional)

Output:

ORDER_ID	ORDER_DATE	COST	CUSTOMER_ID	CUSTOMER_ID	
1	2023-12-11	5000	1	1	Clark ALEX
2	2023-12-11	6000	1	1	Clark ALEX
3	2023-12-11	5000	3	3	Sherif Cairo
4	2023-12-11	5000	2	2	Ibrahim ALEX
5	2023-12-11	5000	4	4	Mohamed Cairo



Left joins

queries.sql + 3zw47atvv NEW MYSQL RUN

```
  ADDRESS VARCHAR(500) NOT NULL
);

CREATE TABLE ORDERS
(
  ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
  ORDER_DATE DATE NOT NULL,
  COST INT NOT NULL,
  CUSTOMER_ID INT NOT NULL,
  foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
);
-- insert
INSERT INTO CUSTOMERS (NAME, ADDRESS)
VALUES ('Clark', 'ALEX'),
       ('Ibrahim', 'ALEX'),
       ('Sherif', 'Cairo'),
       ('Mohamed', 'Cairo'),
       ('ALI', 'Cairo'),
       ('Ahmed', 'Cairo'),
       ('Ava', 'aswan'),
       ('Salma', 'aswan');

INSERT INTO ORDERS (ORDER_DATE, COST, CUSTOMER_ID)
VALUES
  ('2023-12-11', 5000, 1),
  ('2023-12-11', 6000, 1),
  ('2023-12-11', 5000, 3),
  ('2023-12-11', 5000, 2),
  ('2023-12-11', 5000, 4);

SELECT * FROM Customers
LEFT JOIN ORDERS ON ORDERS.Customer_ID = Customers.Customer_ID;
```

STDIN

Input for the program (Optional)

Output:

CUSTOMER_ID	NAME	ADDRESS	ORDER_ID	ORDER_DATE	COST
1	Clark	ALEX	1	2023-12-11	5000	1
1	Clark	ALEX	2	2023-12-11	6000	1
2	Ibrahim	ALEX	4	2023-12-11	5000	2
3	Sherif	Cairo	3	2023-12-11	5000	3
4	Mohamed	Cairo	5	2023-12-11	5000	4
5	ALI	Cairo	NULL	NULL	NULL	NULL
6	Ahmed	Cairo	NULL	NULL	NULL	NULL
7	Ava	aswan	NULL	NULL	NULL	NULL
8	Salma	aswan	NULL	NULL	NULL	NULL

Right join

queries.sql + 3zw47atvv NEW MYSQL RUN

```
  ADDRESS VARCHAR(500) NOT NULL
);

CREATE TABLE ORDERS
(
  ORDER_ID INT AUTO_INCREMENT PRIMARY KEY,
  ORDER_DATE DATE NOT NULL,
  COST INT NOT NULL,
  CUSTOMER_ID INT NOT NULL,
  foreign KEY (CUSTOMER_ID) references CUSTOMERS(CUSTOMER_ID)
);
-- insert
INSERT INTO CUSTOMERS (NAME, ADDRESS)
VALUES ('Clark', 'ALEX'),
       ('Ibrahim', 'ALEX'),
       ('Sherif', 'Cairo'),
       ('Mohamed', 'Cairo'),
       ('ALI', 'Cairo'),
       ('Ahmed', 'Cairo'),
       ('Ava', 'aswan'),
       ('Salma', 'aswan');

INSERT INTO ORDERS (ORDER_DATE, COST, CUSTOMER_ID)
VALUES
  ('2023-12-11', 5000, 1),
  ('2023-12-11', 6000, 1),
  ('2023-12-11', 5000, 3),
  ('2023-12-11', 5000, 2),
  ('2023-12-11', 5000, 4);

SELECT * FROM Customers
right JOIN ORDERS ON ORDERS.Customer_ID = Customers.Customer_ID;
```

STDIN

Input for the program (Optional)

Output:

CUSTOMER_ID	NAME	ADDRESS	ORDER_ID	ORDER_DATE	COST
1	Clark	ALEX	1	2023-12-11	5000	1
1	Clark	ALEX	2	2023-12-11	6000	1
3	Sherif	Cairo	3	2023-12-11	5000	3
2	Ibrahim	ALEX	4	2023-12-11	5000	2
4	Mohamed	Cairo	5	2023-12-11	5000	4


Thank
you