Kushari Al Rahmani DATABASE

BY:

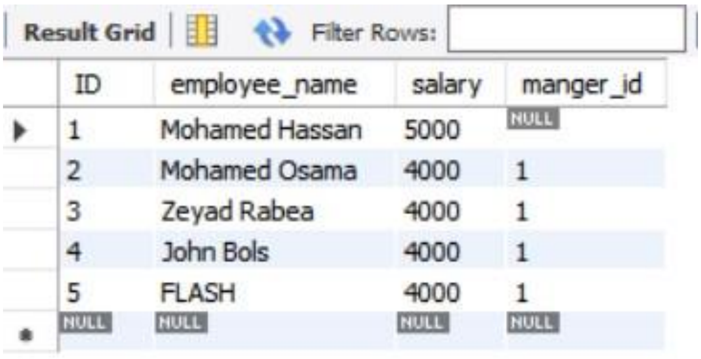
زياد ربيع محمد 20211513

حسن محمد حسن 20211512

جون بولس جاد الرب 20211511

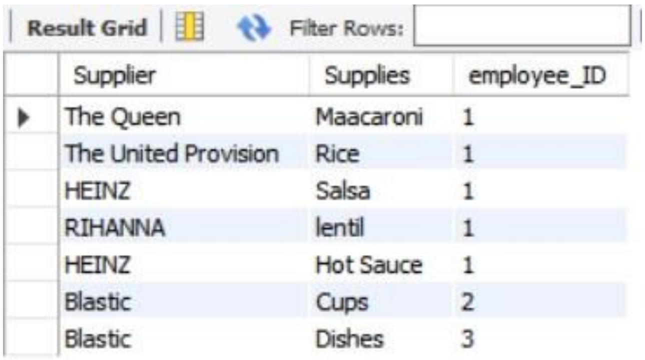
رياضة وحاسب

Employee Table:



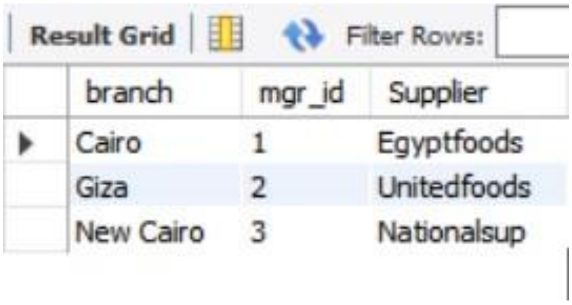
ID	employee_name	salary	manger_id
1	Mohamed Hassan	5000	NULL
2	Mohamed Osama	4000	1
3	Zeyad Rabea	4000	1
4	John Bols	4000	1
5	FLASH	4000	1
NULL	NULL	NULL	NULL

Suppliers Table:



Supplier	Supplies	employee_ID
The Queen	Maacaroni	1
The United Provision	Rice	1
HEINZ	Salsa	1
RIHANNA	lentil	1
HEINZ	Hot Sauce	1
Blastic	Cups	2
Blastic	Dishes	3

Branches Table:



branch	mgr_id	Supplier
Cairo	1	Egyptfoods
Giza	2	Unitedfoods
New Cairo	3	Nationalsup

QUERIES

--Creating Database for the facility

```
CREATE database elrahmany;
```

--Creating The Table of employee

```
CREATE TABLE `employee) `

`  ID` int NOT NULL,

`  employee_name` varchar(40) DEFAULT NULL,

`  salary` int DEFAULT NULL,

`  manger_id` int DEFAULT NULL,

  PRIMARY KEY (`ID`),

  KEY `manger_id` (`manger_id`),

  CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`manger_id`) REFERENCES `employee` (`ID`));
```

--Dumping data for table `employee`

```
INSERT INTO `employee` VALUES (1,'Mohamed Hassan',5000,NULL),(2,'Mohamed Osama',4000,1),(3,'Zeyad Rabea',4000,3),(4,'John Bols',4000,2),(5,'FLASH',4000,4);
```

-- Creating The Table Of Suppliers that bring provisions

```
CREATE TABLE `suppliers) `
`  Supplier` varchar(40) DEFAULT NULL,
`  Supplies` varchar(40) DEFAULT NULL,
`  employee_ID` int DEFAULT NULL,
  KEY `employee_ID` (`employee_ID`),
  CONSTRAINT `suppliers_ibfk_1` FOREIGN KEY (`employee_ID`) REFERENCES `employee` (`ID`));
```

--Dumping data for table `suppliers`

```
INSERT INTO `suppliers`    VALUES ('The Queen','Maacaroni',1),('The United
Provision','Rice',2),('HEINZ','Salsa',3),('RIHANNA','lentil',4),('HEINZ','Hot
Sauce',5),('Blastic','Cups',2),('Blastic','Dishes',3);
```

-- Creating The Table Of Branches

```
 CREATE TABLE `Branches ) `
` branch` varchar,(40)
` mgr_id` INT,
` Supplier` varchar(20),
 PRIMARY KEY (mgr_id),
 FOREIGN KEY (mgr_id) REFERENCES `employee`(manger_id);(
```

-- Dumping data for table `Branchess`

```
INSERT INTO `Branches`   VALUES ('Cairo', 1,'Egyptfoods'),('Giza',2,'Unitedfoods'),('New Cairo',3,'Nationalsup');
```

--To Show us What We have inserted in the table

```
 SELECT * FROM employee;
```

-- To show us only the different values

```
 SELECT DISTINCT `mgr_id` FROM `employee ;`
```

-- Knowing how much employees take in total salary

```
 SELECT SUM(salary)  FROM employee;
```

-- Used To Insert certain things in both tables

```
 INSERT INTO suppliers VALUES('Blastic','Dishes',3);
```

-- Used to modify COLUMNS in table

```
ALTER TABLE table_n
ADD column_n;
```

-- Used to select only the data that matches the condition

```
SELECT * FROM branches
WHERE mgr_id=1;
```

--Used to select only the data that matches the condition (AND OR) multiple conditions

```
SELECT * FROM employee
WHERE salary > 2000 OR ID=3;
```

--Used to sort columns by specific order(s) (ascending or descending)

```
SELECT COUNT(ID) FROM employee
ORDER BY COUNT(ID) ASC;
```

--Used to test for empty values

```
SELECT * FROM branches
WHERE `mgr_id` IS NOT NULL;
```

--Used to modify existing record in table

```
UPDATE employee
SET salary= 3000
WHERE ID=4;
```

--Used to delete (entire/specific) records in table

```
DELETE FROM suppliers WHERE employee_ID=4;
```

--Used to specify number of records to return

```
SELECT TOP 5 FROM employee;
```

--Used to return smallest/largest value of selected column

```
SELECT MIN(ID) FROM employee;
```

--Used to search for a specific pattern in a column

```
SELECT * FROM employee
WHERE employee_name LIKE "a%" OR "_a;"_
```

--Used to specify multiple values in a WHERE clause

```
SELECT * FROM suppliers
WHERE employee_ID IN (SELECT * FROM employee);
```

--Used to select values between given ranges

```
SELECT * FROM employee
WHERE ID BETWEEN 2 AND 6;
```

--Used to give a table, column a temporary name

```
SELECT suppliers.employee_ID AS ID
FROM suppliers;
```

--Used to select records that have matching values in both tables

```
SELECT ID
FROM employee
INNER JOIN ID
ON employee.ID = suppliers.employee_ID;
```

--Used to combine multiple select statments

```
SELECT * FROM employee
UNION
SELECT * FROM branches;
```

--Used to group rows that has the same values

```
SELECT COUNT(mgr_id), branch  FROM branches
GROUP BY branch
ORDER BY COUNT(mgr_id) DESC;
```

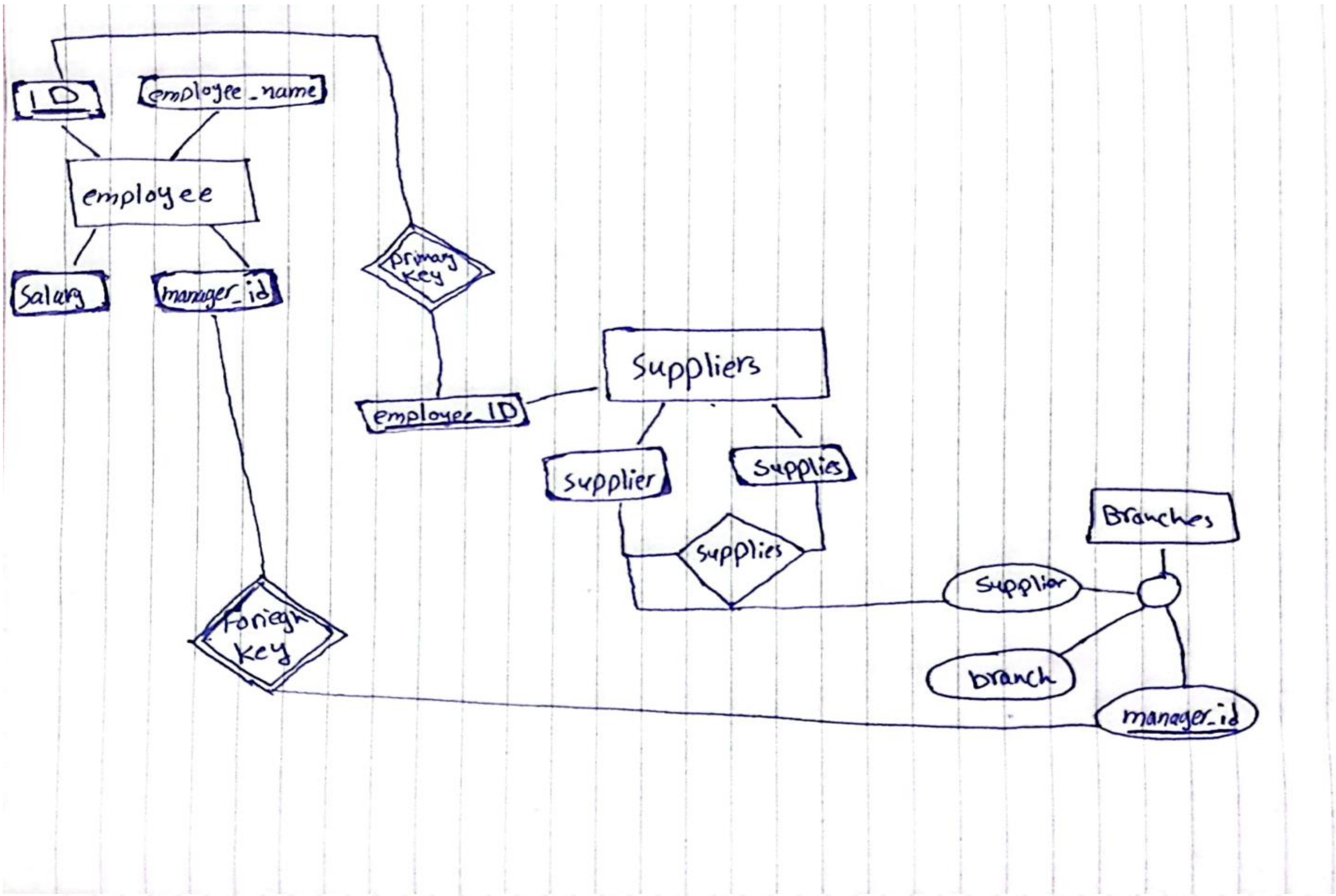
--Used to specify a condition when WHERE connot be used

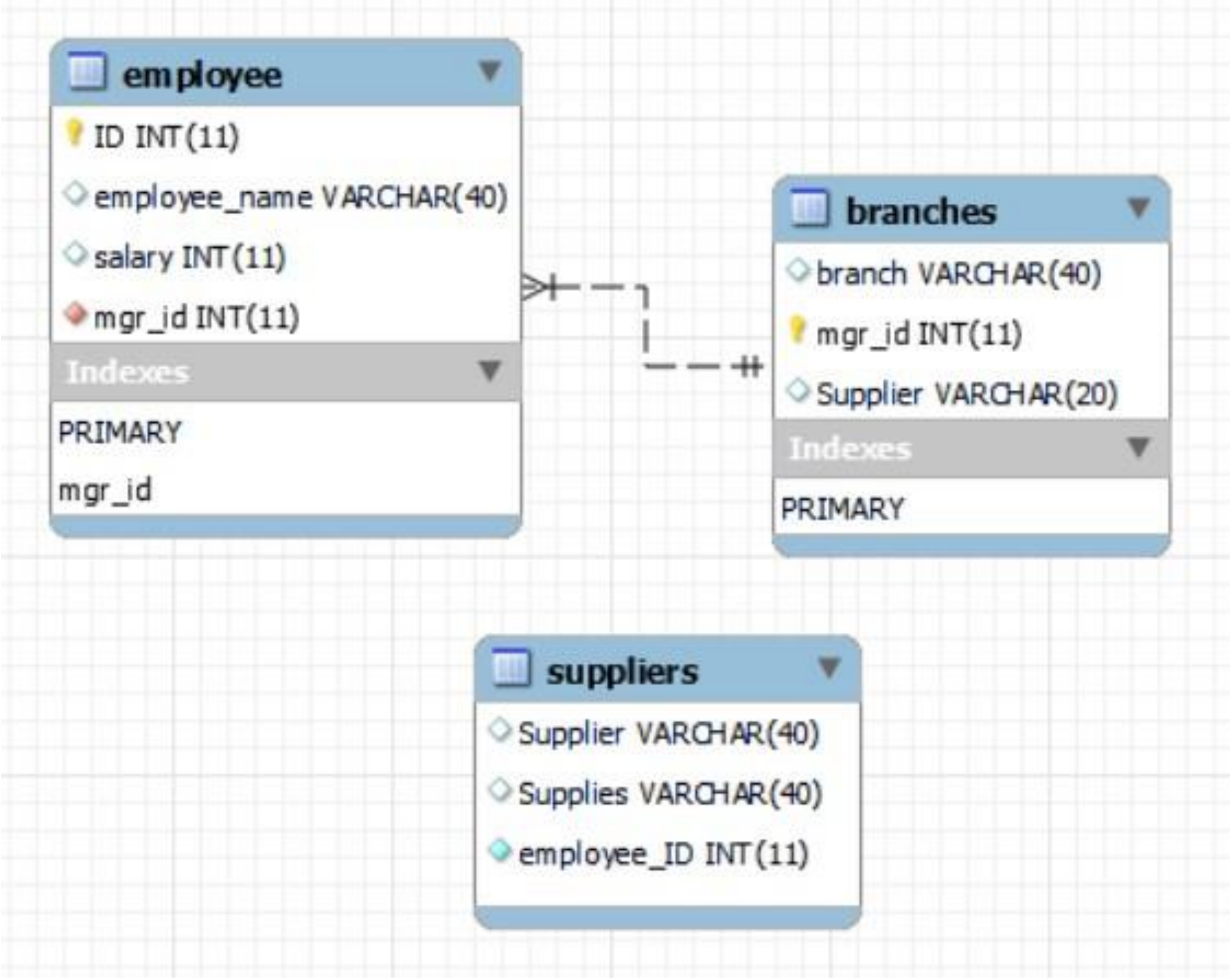
```
SELECT COUNT(ID), employee_name
FROM employee
HAVING COUNT(ID) > 1;
```

--Used to test existence of any record

```
SELECT * FROM employee
WHERE EXISTS
(SELECT * FROM supliers WHERE employee_ID >1);
```

ER Design





employee

ID INT(11)

employee_name VARCHAR(40)

salary INT(11)

mgr_id INT(11)

Indexes

PRIMARY

mgr_id

branches

branch VARCHAR(40)

mgr_id INT(11)

Supplier VARCHAR(20)

Indexes

PRIMARY

suppliers

Supplier VARCHAR(40)

Supplies VARCHAR(40)

employee_ID INT(11)