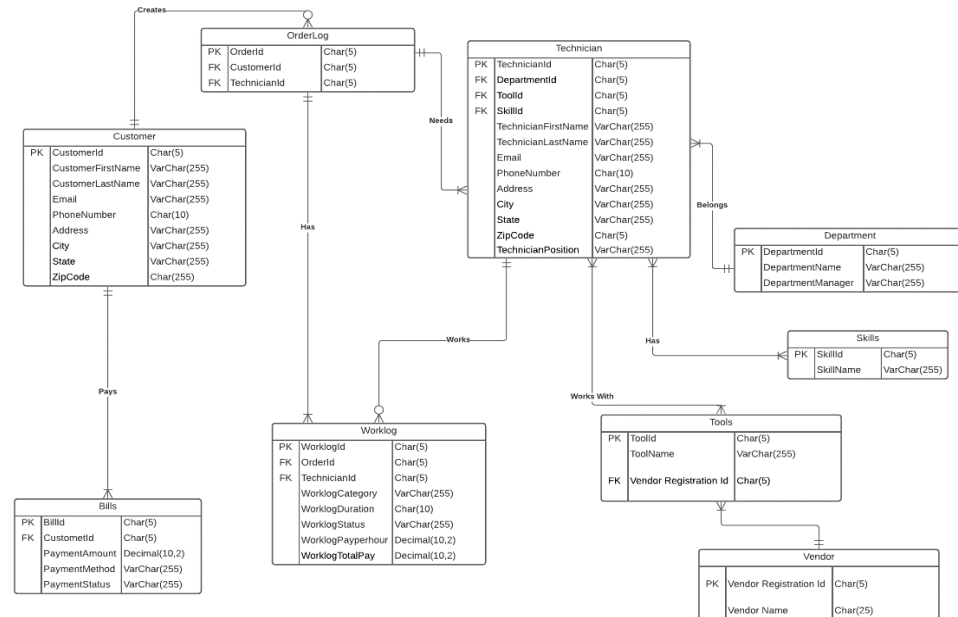


Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

IPA 3

ERD Used



SQL: I have considered the ERD before normalization and written SQL queries using it.

Note: All queries are attached in dbo file

Part 1: Creating Database

```
create database ipafinal;
use ipafinal
go
```

Part 2: Creating tables and Inserting values

```
--- Create Vendor Table
CREATE TABLE vendor
(
    vendor_id    CHAR(5)      NOT NULL,
    vendor_name  VARCHAR(25)  NOT NULL,
    CONSTRAINT pk_vendor PRIMARY KEY (vendor_id)
);
--- Insert values into vendors
INSERT INTO vendor VALUES('V001','WestReg Tooling supply');
INSERT INTO vendor VALUES('V002','Helly Tooling supply');
INSERT INTO vendor VALUES('V003','Ria Tools');
```

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

----- Create Tools table

```
CREATE TABLE tool
(
tool_id      CHAR(5)      NOT NULL,
tool_name    VARCHAR(255) NOT NULL,
vendor_id    CHAR(5)      NOT NULL,
CONSTRAINT pk_tool PRIMARY KEY (tool_id),
CONSTRAINT fk_tool_vendor FOREIGN KEY (vendor_id) REFERENCES vendor(vendor_id)
);
```

----- Insert Values into Tools Table

```
INSERT INTO tool VALUES('TK001','Shackle','V001');
INSERT INTO tool VALUES('TK002','Rod','V002');
INSERT INTO tool VALUES('TK003','Compass','V002');
INSERT INTO tool VALUES('TK004','Twister','V001');
INSERT INTO tool VALUES('TK005','Screwdriver','V003');
INSERT INTO tool VALUES('TK006','Mop','V001');
```

----- Create Skills table

```
CREATE TABLE skills
(
skill_id     CHAR(5)      NOT NULL,
skill_name   VARCHAR(255) NOT NULL,
CONSTRAINT pk_skill PRIMARY KEY (skill_id)
);
```

----- Insert Values into skills

```
INSERT INTO skills VALUES('SK001','Electrician');
INSERT INTO skills VALUES('SK002','Plumbing');
INSERT INTO skills VALUES('SK003','Drawing');
INSERT INTO skills VALUES('SK004','Assembling');
INSERT INTO skills VALUES('SK005','Carpenting');
INSERT INTO skills VALUES('SK006','Interior Designer');
INSERT INTO skills VALUES('SK007','Flooring Expert');
INSERT INTO skills VALUES('SK008','Lock Smith');
```

----- Create Department Table

```
CREATE TABLE department
(
department_id CHAR(5)      NOT NULL,
department_name VARCHAR(255) NOT NULL,
department_manager VARCHAR(255) NOT NULL,
CONSTRAINT pk_department PRIMARY KEY (department_id)
);
```

-----Insert values into department

```
INSERT INTO department VALUES('D001','BasicHome','Chris');
INSERT INTO department VALUES('D002','AdvanceHome','Stella');
INSERT INTO department VALUES('D003','FineHome','Bonnie');
INSERT INTO department VALUES('D004','Interiors','Micheal');
```

----- Create Technician Table

```
CREATE TABLE technician
(
technician_id CHAR(5)      NOT NULL,
department_id CHAR(5)      NOT NULL,
tool_id       CHAR(5)      NOT NULL,
skill_id      CHAR(5)      NOT NULL,
```

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

```
technician_fname VARCHAR(255) ,
technician_lname VARCHAR(255) ,
technician_email VARCHAR(255) ,
technician_phone VARCHAR(10) ,
technician_address VARCHAR(255) ,
technician_city VARCHAR(255) ,
technician_state VARCHAR(255) ,
technician_zip VARCHAR(5) ,
technician_designation varchar(255),
CONSTRAINT pk_technician PRIMARY KEY (technician_id),
CONSTRAINT fk_technician_department FOREIGN KEY (department_id) REFERENCES
department(department_id),
CONSTRAINT fk_technician_tool FOREIGN KEY (tool_id) REFERENCES tool(tool_id),
CONSTRAINT fk_technician_skills FOREIGN KEY (skill_id) REFERENCES skills(skill_id),
);

-----Insert value into department
INSERT INTO technician
VALUES('T001','D001','TK001','SK001','Harry','Potter','hp@gmail.com','6789875432','7431
frankford road','Dallas','Texas','75256','Electrician');
INSERT INTO technician
VALUES('T002','D001','TK002','SK002','Emerald','Jade','ejade@gmail.com','6234567891','567
8 Frisco street','Dallas','Texas','75213','Plumber');
INSERT INTO technician
VALUES('T003','D002','TK003','SK003','Emily','Kane','em@gmail.com','6724536789','6534
estates on back','Plano','Texas','76342','Architect');
INSERT INTO technician
VALUES('T004','D003','TK004','SK004','Lara','Skaut','lsk@gmail.com','7534256789','5342
Mandrian Street','DFW','Texas','76345','Decorator');
INSERT INTO technician
VALUES('T005','D002','TK005','SK005','Dia','Amen','da@gmail.com','5234567890','4213
Estaes on Pearl','Houston','Texas','73498','Floor Planner');
INSERT INTO technician
VALUES('T006','D003','TK004','SK005','Diam','Mehta','dm@gmail.com','5234567111','7575
Estates','California','CA','76788','Floor maintenance');

----- Create Customer Table

CREATE TABLE customer
(
customer_id CHAR(5) NOT NULL,
customer_fname VARCHAR(255) NOT NULL,
customer_lname VARCHAR(255) ,
email VARCHAR(255) ,
phone VARCHAR(12) ,
address VARCHAR(255) ,
city VARCHAR(255) ,
state CHAR(255) ,
zip CHAR(5) ,
CONSTRAINT pk_customer PRIMARY KEY (customer_id),
);

-----Insert Values into customer
INSERT INTO customer VALUES('C001','Jenny','Buchman','jb@gmail.com','718-496-7223','75
West 205 St','Bronx','NY','10468');
INSERT INTO customer VALUES('C002','Wendy','Heydemark','wh@gmail.com','303-986-
7020','2922 Baseline Rd','Boulder','CO','80303');
```

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

```
INSERT INTO customer VALUES('C003','Hallie','Hull','hhull@gmail.com','415-549-4278','3800 Waldo Ave, #14F','San Francisco','CA','94123');
INSERT INTO customer VALUES('C004','Klee','Hull','kh@gmail.com','415-549-4278','3800 Waldo Ave, #14F','San Francisco','CA','94123');
INSERT INTO customer VALUES('C005','Christian','Kells','ck@gmail.com','212-771-4680','114 Horatio St','New York','NY','10014');
INSERT INTO customer VALUES('C006','jane','Kellsey','jk@gmail.com','650-836-7128','390 Serra Mall','Palo Alto','CA','94305');
INSERT INTO customer VALUES('C007','Paddy','O','Furniture','pf@gmail.com','941-925-0752','1442 Main St','Sarasota','FL','34236');
INSERT INTO customer VALUES('C008','Padmini','Helly',NULL,'941-925-0788','1991 Estate St','Sarans','CA','67823');
INSERT INTO customer VALUES('C009','Damini','Malhotra','dm@gmail.com','678-092-0987','7575 Frankford Road','Dallas','TX','75252');
INSERT INTO customer VALUES('C010','Priya','Desai','pd@gmail.com','941-925-1234','1235 nainital street','denvar','DN','34239');
```

---- Create Order Log table

```
CREATE TABLE orderlog
(
    order_id    CHAR(5)      NOT NULL,
    customer_id CHAR(5)      NOT NULL,
    technician_id CHAR(5)    NOT NULL,
    CONSTRAINT pk_orderlog PRIMARY KEY (order_id),
    CONSTRAINT fk_orderlog_customer FOREIGN KEY (customer_id) REFERENCES
customer(customer_id),
    CONSTRAINT fk_orderlog_technician FOREIGN KEY (technician_id) REFERENCES
technician(technician_id)
);
-----insert values into orderlog
INSERT INTO orderlog VALUES('OR001','C001','T001');
INSERT INTO orderlog VALUES('OR002','C002','T002');
INSERT INTO orderlog VALUES('OR003','C001','T003');
INSERT INTO orderlog VALUES('OR004','C002','T003');
INSERT INTO orderlog VALUES('OR005','C002','T003');
INSERT INTO orderlog VALUES('OR006','C003','T004');
INSERT INTO orderlog VALUES('OR007','C004','T005');
INSERT INTO orderlog VALUES('OR007','C005','T001');
```

----- Create Worklog table

```
CREATE TABLE worklog
(
    worklog_id    CHAR(5)      NOT NULL,
    order_id      CHAR(5)      NOT NULL,
    technician_id  CHAR(5)      ,
    worklog_category VARCHAR(255) ,
    worklog_duration DECIMAL(10,2) ,
    woklog_status  VARCHAR(255) NOT NULL,
    worklogpay_perhour DECIMAL(10,2),
    worklogpay_total DECIMAL(10,2),
    CONSTRAINT pk_worklog PRIMARY KEY (worklog_id),
    CONSTRAINT fk_worklog_technician FOREIGN KEY (technician_id) REFERENCES
technician(technician_id)
);
```

-----Insert values into worklog table

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

```
INSERT INTO worklog
VALUES('WL001','OR001','T001','Housing','10.00','active','5.00','50.00');
INSERT INTO worklog
VALUES('WL002','OR002','T002','Housing','15.00','active','5.00','75.00');
INSERT INTO worklog
VALUES('WL003','OR003','T001','Housing','12.00','done','5.00','60.00');
INSERT INTO worklog
VALUES('WL004','OR004','T003','Society','12.00','done','6.00','72.00');
INSERT INTO worklog
VALUES('WL005','OR004','T004','Society','12.00','done','6.00','72.00');
INSERT INTO worklog
VALUES('WL006','OR004','T005','Society','10.00','done','1.00','10.00');
INSERT INTO worklog
VALUES('WL007','OR005','T001','Housing','12.00','done','5.00','60.00');
INSERT INTO worklog
VALUES('WL008','OR006','T002','Housing','12.00','done','6.00','72.00');
```

----Create Bill table

```
CREATE TABLE bill
(
bill_id      CHAR(5)      NOT NULL,
customer_id  CHAR(5)      NOT NULL,
payment_amount DECIMAL(10,2),
payment_method VARCHAR(255),
payment_status VARCHAR(255),
CONSTRAINT pk_bill PRIMARY KEY (bill_id),
CONSTRAINT fk_bill_customer FOREIGN KEY (customer_id) REFERENCES customer(customer_id)
);
```

--- Insert values into bill table

```
INSERT INTO bill VALUES('B001','C002','120.00','Cash','Complete');
INSERT INTO bill VALUES('B002','C002','12.00','Cash','Complete');
INSERT INTO bill VALUES('B003','C001','10.00',NULL,'Pending');
INSERT INTO bill VALUES('B004','C003','20.00',NULL,'Pending');
INSERT INTO bill VALUES('B005','C004','5.00',NULL,'Pending');
```

Part 3: Viewing the tables

A. Customer Table

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables like 'customer', 'technician', and 'orderlog'. The main query window contains the following SQL script:

```
INSERT INTO bill VALUES ('B001','C002','120.00','Cash','Complete');
INSERT INTO bill VALUES ('B002','C002','12.00','Cash','Complete');
INSERT INTO bill VALUES ('B003','C001','10.00',NULL,'Pending');
INSERT INTO bill VALUES ('B004','C003','20.00',NULL,'Pending');
INSERT INTO bill VALUES ('B005','C004','5.00',NULL,'Pending');

-- Selecting Customer table
select * from customer;

-- Selecting values from Technician table
select * from technician;

-- Selecting values from orderlog table
select * from orderlog;
```

The Results pane shows the output of the 'select * from customer;' query, displaying 10 rows of customer data:

customer_id	customer_fname	customer_lname	email	phone	address	city	state	zip
C001	Jenny	Buchman	jb@gmail.com	718-406-7223	75 West 205 St	Bronx	NY	10468
C002	Wendy	Heydemark	wh@gmail.com	303-986-7020	2922 Baseline Rd	Boulder	CO	80303
C003	Halle	Hull	hhull@gmail.com	415-549-4278	3800 Waldo Ave, #14F	San Francisco	CA	94123
C004	Klee	Hull	kh@gmail.com	415-549-4278	3800 Waldo Ave, #14F	San Francisco	CA	94123
C005	Christian	Kells	ck@gmail.com	212-771-4680	114 Horatio St	New York	NY	10014
C006	jane	Kellsey	jk@gmail.com	650-836-7128	390 Serra Mall	Palo Alto	CA	94305
C007	Paddy	O'Furniture	pf@gmail.com	941-925-0752	1442 Main St	Sarasota	FL	34236
C008	Padmini	Helly	NULL	941-925-0788	1991 Estate St	Sarasota	CA	67823
C009	Damini	Malhotra	dm@gmail.com	678-092-0987	7575 Frankford Road	Dallas	TX	75252
C010	Priya	Desai	pd@gmail.com	941-925-1234	1235 nainital street	denvar	DN	34239

The status bar at the bottom indicates 'Query executed successfully.' and '10 rows'.

B. Technician Table

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables like 'technician', 'orderlog', and 'bills'. The main query window contains the following SQL script:

```
select * from customer;

-- Selecting values from Technician table
select * from technician;

-- Selecting values from orderlog table
select * from orderlog;

-- Selecting values from bills table
select * from bill;

-- Selecting values from skills table
```

The Results pane shows the output of the 'select * from technician;' query, displaying 10 rows of technician data:

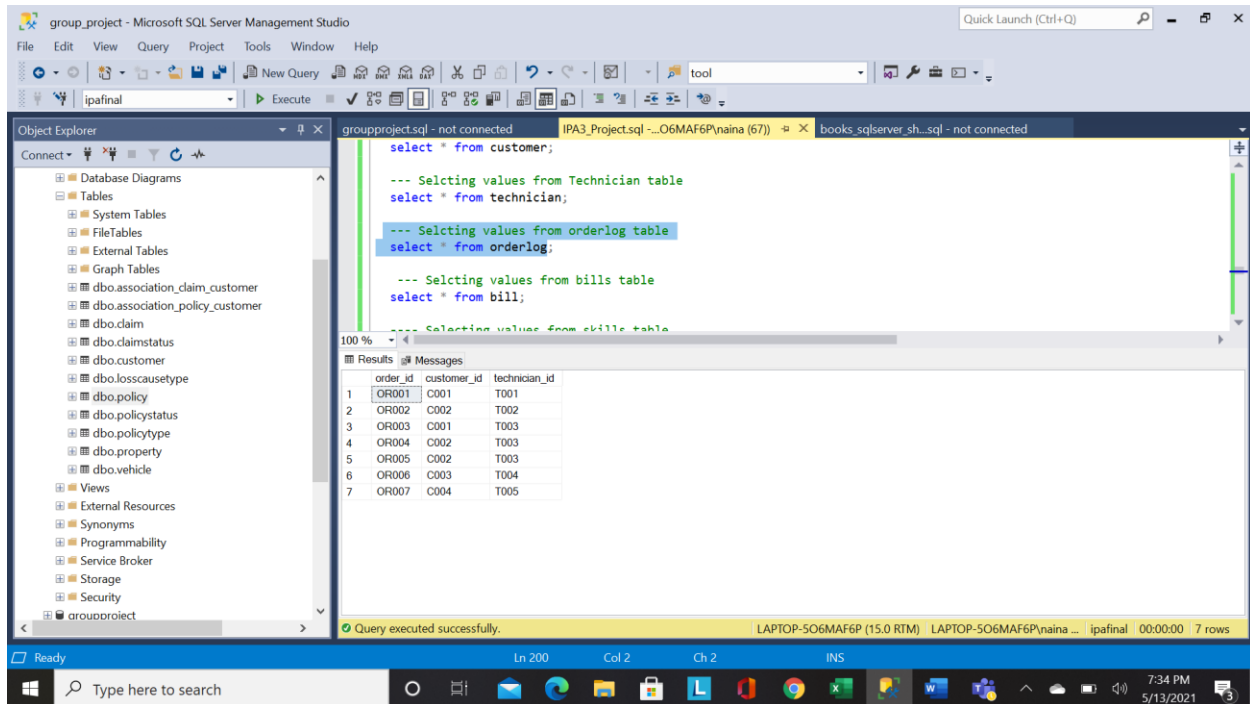
technician_id	department_id	tool_id	skill_id	technician_fname	technician_lname	technician_email	technician_phone	technician_address	technician_city	technician_state
T001	D001	TK001	SK001	Harry	Potter	hp@gmail.com	6789875432	7431 frankford road	Dallas	Texas
T002	D001	TK002	SK002	Emerald	Jade	ejade@gmail.com	6234567891	5678 Frisco street	Dallas	Texas
T003	D002	TK003	SK003	Emily	Kane	em@gmail.com	6724536789	6534 estates on back	Plano	Texas
T004	D003	TK004	SK004	Lara	Skaut	lsk@gmail.com	7534567890	5342 Mandrian Street	DFW	Texas
T005	D002	TK005	SK005	Dia	Amen	da@gmail.com	5234567890	4213 Estates on Pearl	Houston	Texas
T006	D003	TK004	SK005	Diam	Mehta	dm@gmail.com	5234567111	7575 Estates	California	CA
T007	D002	TK004	SK004	Dhia	Verma	NULL	5234567222	7576 Estates	California	CA
T008	D002	TK004	SK004	TIA	Verma	NULL	5234582222	pearl 71 Estates	California	CA
T009	D001	TK004	SK004	Sahira	Verma	NULL	5234567222	7576 Estates	California	CA
T010	D003	TK004	SK004	Karan	Gill	NULL	5234582222	pearl 71 Estates	California	CA

The status bar at the bottom indicates 'Query executed successfully.' and '10 rows'.

C. Order log table

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables like dbo.association_claim_customer, dbo.association_policy_customer, dbo.claim, dbo.claimstatus, dbo.customer, dbo.losscausetype, dbo.policy, dbo.policystatus, dbo.policytype, dbo.property, and dbo.vehicle. The Query window on the right contains the following SQL code:

```
select * from customer;

--- Selecting values from Technician table
select * from technician;

--- Selecting values from orderlog table
select * from orderlog;

--- Selecting values from bills table
select * from bill;

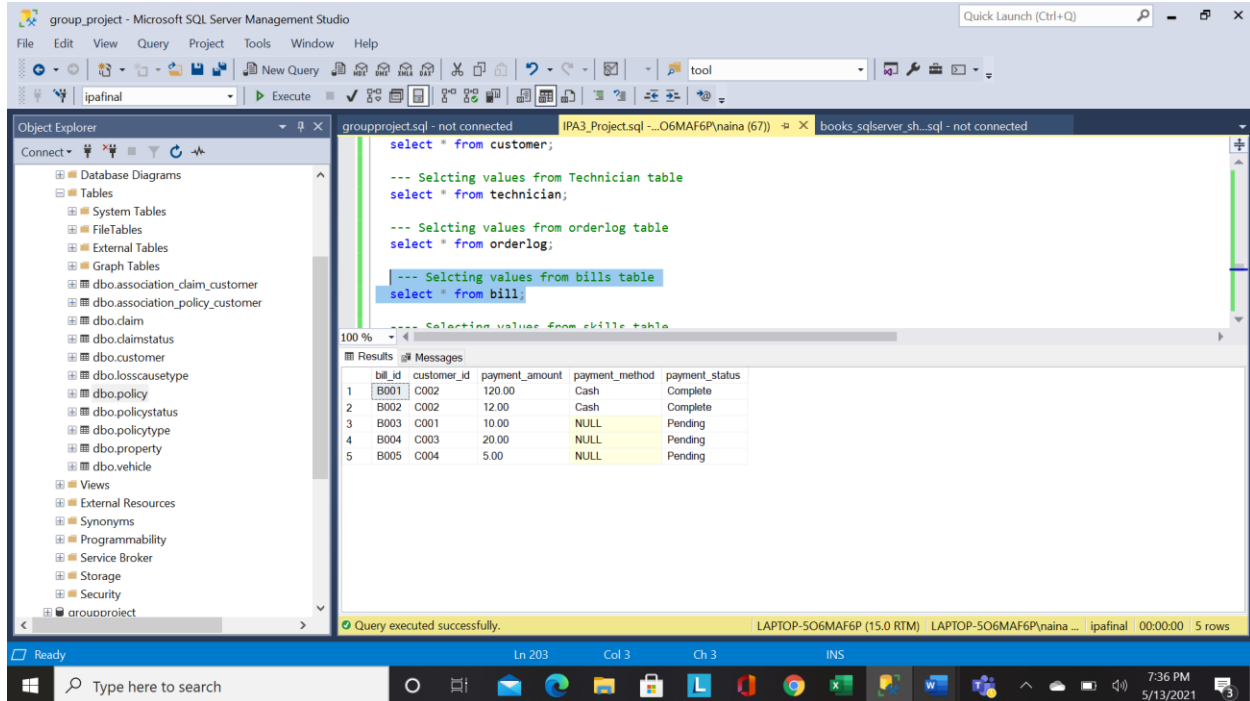
--- Selecting values from skills table
```

The Results pane shows the output of the query, displaying 7 rows of data with columns order_id, customer_id, and technician_id:

	order_id	customer_id	technician_id
1	OR001	C001	T001
2	OR002	C002	T002
3	OR003	C001	T003
4	OR004	C002	T003
5	OR005	C002	T003
6	OR006	C003	T004
7	OR007	C004	T005

The status bar at the bottom indicates "Query executed successfully." and "7 rows".

D. Bills Table



The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables like dbo.association_claim_customer, dbo.association_policy_customer, dbo.claim, dbo.claimstatus, dbo.customer, dbo.losscausetype, dbo.policy, dbo.policystatus, dbo.policytype, dbo.property, and dbo.vehicle. The Query window on the right contains the following SQL code:

```
select * from customer;

--- Selecting values from Technician table
select * from technician;

--- Selecting values from orderlog table
select * from orderlog;

--- Selecting values from bills table
select * from bill;

--- Selecting values from skills table
```

The Results pane shows the output of the query, displaying 5 rows of data with columns bill_id, customer_id, payment_amount, payment_method, and payment_status:

	bill_id	customer_id	payment_amount	payment_method	payment_status
1	B001	C002	120.00	Cash	Complete
2	B002	C002	12.00	Cash	Complete
3	B003	C001	10.00	NULL	Pending
4	B004	C003	20.00	NULL	Pending
5	B005	C004	5.00	NULL	Pending

The status bar at the bottom indicates "Query executed successfully." and "5 rows".

E. Skills Table

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables, views, and security. The main query window shows a SQL script with four queries: selecting from orderlog, bills, skills, and tools tables. The results pane at the bottom displays the output of the third query, which selects from the skills table. The results are as follows:

skill_id	skill_name
SK001	Electrician
SK002	Plumbing
SK003	Drawing
SK004	Assembling
SK005	Carpentery
SK006	Interior Designer
SK007	Flooring Expert
SK008	Lock Smith

F. Tool Table

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables, views, and security. The main query window shows a SQL script with four queries: selecting from orderlog, bills, skills, and tools tables. The results pane at the bottom displays the output of the fourth query, which selects from the tools table. The results are as follows:

tool_id	tool_name	vendor_id
TK001	Shackle	V001
TK002	Rod	V002
TK003	Compass	V002
TK004	Twister	V001
TK005	Screwdriver	V003
TK006	Mop	V001

G. Vendor Table

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'groupproject'. The query window on the right contains the following SQL code:

```
---- Selecting values from skills table
select * from skills;

---- Selecting values from tools table
select * from tool;

---- Selecting values from vendor table
select * from vendor;

---- Selecting values from department table
```

The Results pane shows the output of the 'select * from vendor;' query:

vendor_id	vendor_name
V001	WestReg Tooling supply
V002	Helly Tooling supply
V003	Ria Tools

The status bar at the bottom indicates 'Query executed successfully.' and '3 rows'.

H. Department Table

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'groupproject'. The query window on the right contains the following SQL code:

```
---- Selecting values from tools table
select * from tool;

---- Selecting values from vendor table
select * from vendor;

---- Selecting values from department table
select * from department;

--Selecting values from worklog
```

The Results pane shows the output of the 'select * from department;' query:

department_id	department_name	department_manager
D001	BasicHome	Chris
D002	AdvanceHome	Stella
D003	FineHome	Bonnie
D004	Interiors	Micheal

The status bar at the bottom indicates 'Query executed successfully.' and '4 rows'.

I. Worklog Table

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables like 'vendor', 'department', and 'worklog'. The central query editor contains the following SQL code:

```
---- Selecting values from vendor table
select * from vendor;

---- Selecting values from department table
select * from department;

--Selecting values from worklog
select * from worklog;

-- INNER JOIN--TECHNICIAN AND WORKLOG
```

The Results pane at the bottom displays the output of the query, showing 8 rows of data. The columns are: worklog_id, order_id, technician_id, worklog_category, worklog_duration, worklog_status, worklogpay_perhour, and worklogpay_total.

worklog_id	order_id	technician_id	worklog_category	worklog_duration	worklog_status	worklogpay_perhour	worklogpay_total
WL001	OR001	T001	Housing	10.00	active	5.00	50.00
WL002	OR002	T002	Housing	15.00	active	5.00	75.00
WL003	OR003	T001	Housing	12.00	done	5.00	60.00
WL004	OR004	T003	Society	12.00	done	6.00	72.00
WL005	OR004	T004	Society	12.00	done	6.00	72.00
WL006	OR004	T005	Society	10.00	done	1.00	10.00
WL007	OR005	T001	Housing	12.00	done	5.00	60.00
WL008	OR006	T002	Housing	12.00	done	6.00	72.00

The status bar at the bottom indicates that the query was executed successfully and returned 8 rows.

Joins

1. INNER JOIN

Ans: The inner join is taken on technician and worklog, it is used to know how many hours a technician has worked on which order.

Name : Naina Chowdhary Vallurupalli
Net Id: ncv200001@utdallas.edu

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'groupproject'. The Query Editor in the center contains the following SQL query:

```
--Selecting values from worklog
select * from worklog;

-- INNER JOIN--TECHNICIAN AND WORKLOG
select e1.technician_id, e1.technician_fname, e2.worklog_duration, e2.order_id
from technician e1
inner join worklog e2
on e1.technician_id=e2.technician_id;

----- FULL OUTER JOIN --- bill and customer
```

The Results pane shows the output of the query, displaying 8 rows of data:

technician_id	technician_fname	worklog_duration	order_id
T001	Harry	10.00	OR001
T002	Emerald	15.00	OR002
T001	Harry	12.00	OR003
T003	Emily	12.00	OR004
T004	Lara	12.00	OR004
T005	Dia	10.00	OR004
T001	Harry	12.00	OR005
T002	Emerald	12.00	OR006

The status bar at the bottom indicates 'Query executed successfully.' and '8 rows'.

2. Full Outer Join

Ans: We are using this join to find out bills of all the customers which are completed, pending as well as customers who do not have any bills.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'groupproject'. The Query Editor in the center contains the following SQL query:

```
----- FULL OUTER JOIN --- bill and customer
select e1.customer_id, e1.customer_fname, e2.bill_id, e2.payment_amount, e2.payment_status
from customer e1
FULL OUTER JOIN bill e2
on e1.customer_id=e2.customer_id;

----- LEFT JOIN --- customers and technicians
```

The Results pane shows the output of the query, displaying 11 rows of data:

customer_id	customer_fname	bill_id	payment_amount	payment_status
C001	Jenny	B003	10.00	Pending
C002	Wendy	B001	120.00	Complete
C002	Wendy	B002	12.00	Complete
C003	Halle	B004	20.00	Pending
C004	Klee	B005	5.00	Pending
C005	Christian	NULL	NULL	NULL
C006	jane	NULL	NULL	NULL
C007	Paddy	NULL	NULL	NULL
C008	Padmini	NULL	NULL	NULL
C009	Damini	NULL	NULL	NULL
C010	Priya	NULL	NULL	NULL

The status bar at the bottom indicates 'Query executed successfully.' and '11 rows'.

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

3. Left Join

Ans: We are using a left join between customer and orderlog. It gives us all customers and any orders that they might have placed.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables like customer, technician, and orderlog. The Query window in the center contains the following SQL code:

```
----- LEFT JOIN --- customers and technicians
select e1.customer_id, e1.customer_fname, e2.technician_id, e2.order_id
from customer e1
LEFT OUTER JOIN orderlog e2
on e1.customer_id=e2.customer_id;

-- RIGHT JOIN--- technician and skills
select e1.technician_id, e1.technician_fname, e2.skill_id, e2.skill_name
```

The Results pane at the bottom displays the output of the first query, showing a list of customers and their associated orders. The data is as follows:

customer_id	customer_fname	technician_id	order_id
C001	Jenny	T001	OR001
C001	Jenny	T003	OR003
C002	Wendy	T002	OR002
C002	Wendy	T003	OR004
C002	Wendy	T003	OR005
C003	Halle	T004	OR006
C004	Klee	T005	OR007
C005	Christian	NULL	NULL
C006	jane	NULL	NULL
C007	Paddy	NULL	NULL
C008	Padmini	NULL	NULL
C009	Damini	NULL	NULL
C010	Priya	NULL	NULL

The status bar at the bottom indicates that the query was executed successfully, returning 13 rows.

4. Right Join

Ans: The table technician and skills are joined using a right join. It tells us that the skills used by technicians. It also gives the skills for which there are no technicians. This can help the handy company to recruit the personal with the specific skill set.

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure, including tables like 'customer', 'technician', and 'skills'. The Query Editor in the center contains the following SQL query:

```
select e1.customer_id, e1.customer_fname, e2.technician_id, e2.order_id
from customer e1
LEFT OUTER JOIN orderlog e2
on e1.customer_id=e2.customer_id;

-- RIGHT JOIN--- technician and skills
select e1.technician_id, e1.technician_fname, e2.skill_id, e2.skill_name
from technician e1
RIGHT OUTER JOIN skills e2
on e1.skill_id=e2.skill_id;

--- Intersect ---technician and worklog
```

The Results pane at the bottom shows the output of the query, which is a table with 13 rows and 4 columns: technician_id, technician_fname, skill_id, and skill_name. The data is as follows:

technician_id	technician_fname	skill_id	skill_name
T001	Harry	SK001	Electrician
T002	Emerald	SK002	Plumbing
T003	Emily	SK003	Drawing
T004	Lara	SK004	Assembling
T007	Dhira	SK004	Assembling
T008	TIA	SK004	Assembling
T009	Sahira	SK004	Assembling
T010	Karan	SK004	Assembling
T005	Dia	SK005	Carpenting
T006	Diam	SK005	Carpenting
NULL	NULL	SK006	Interior Designer
NULL	NULL	SK007	Flooring Expert
NULL	NULL	SK008	Lock Smith

The status bar at the bottom indicates that the query was executed successfully and returned 13 rows.

5. Intersect Join

Ans: Intersect is used here to select technicians who have worked on some worklog. This can help to allocate work to other technicians too.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure. The Query Editor in the center contains the following SQL query:

```
--- Intersect ---technician and worklog
select technician_id from technician
intersect
select technician_id from worklog;

---- Union-- technician and customer
Select e1.customer_id, e1.customer_fname, e1.address, e1.state
from customer e1
Right JOIN orderlog e2
ON e1.customer_id= e2.customer_id WHERE e2.technician_id='T003'
```

The Results pane at the bottom shows the output of the query, which is a table with 5 rows and 1 column: technician_id. The data is as follows:

technician_id
T001
T002
T003
T004
T005

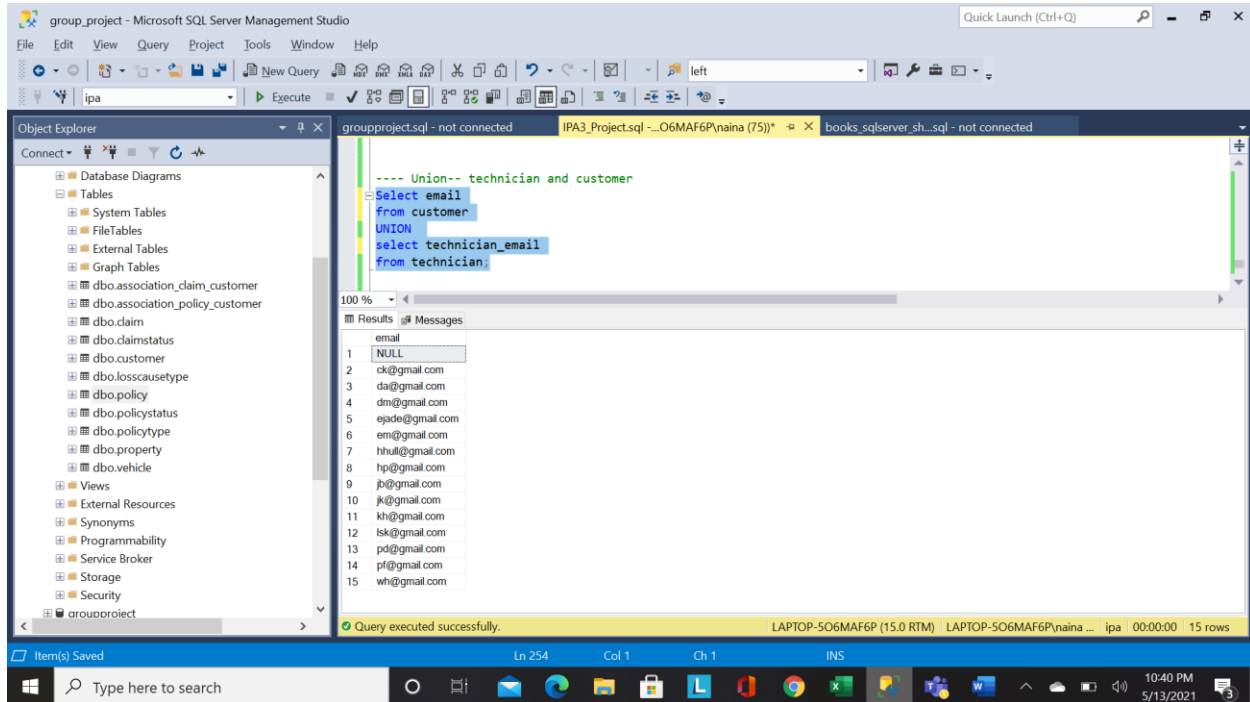
The status bar at the bottom indicates that the query was executed successfully and returned 5 rows.

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

6. Union Join

Ans: This Union join is used here to fetch the email address of all customers and technicians. This can be used to send any combined mails like festival greetings to their Emails.



7. Except Join

Ans: Except Join is used here to fetch the details of Technicians who have not been associated with any worklog. These are basically those who have not been assigned any work till now. This would help the company to assigned work to new technicians.

Name : Naina Chowdhary Vallurupalli

Net Id: ncv200001@utdallas.edu

The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The 'Object Explorer' on the left shows the database structure for 'groupproject'. The 'Query Editor' in the center contains a SQL query that uses an EXCEPT operator to find technician IDs in the 'technician' table that are not in the 'orderlog' table. The query is as follows:

```
select technician_id, technician_name, technician_address, technician_state
from technician WHERE technician_id='T003';

--- Except --- Technician and Orderlog

select technician_id from technician
EXCEPT
select technician_id from orderlog;
```

The 'Results' pane at the bottom shows the output of the query, which is a list of technician IDs. The status bar at the bottom indicates that the query was executed successfully and returned 5 rows.

technician_id
T006
T007
T008
T009
T010