

SQLQuery1.sql - a...UAD\mbrewer7 (97)* X

```
/*Basic Test #2. A similar two INSERT statements executed together in the opposite order should fail. */
select *
from EMPLOYEES
where EmpID = '0222269813';
```

100 %

Results Messages

	EmpID	EmpFName	EmpLName	EmpAddress	EmpPhone#	EmpJobType	EmgContactName	EmgRelationship	EmgContactPhone
1	0222269813	Lovin	Mc	89021 North Jackson Street, Nashville, TN 90821	353-243-5081	T	Savannah James	Wife	808-597-9280

3. Finding out certifications for a given Installer/Technician - NT

The screenshot shows a SQL query being run in SSMS. The query is designed to find certifications for a specific technician based on their Employee ID (EmpID). The results show one row with a certificate number of 97767240, specialty in Landscaping, and a renewal date of 2020-11-28.

```
BasicTest3.sql - ac...SUAD\itsanche (65)* X
/* Basic Test #3
For a given installer/technician, we should be able to find out what certification
he has, and when they need to be renewed */

SELECT Certificate#, TradeSpecialty, DateOfRenewal
FROM TECHNICIANS
WHERE EmpID = '0233456789';

100 %
Results Messages
Certificate# TradeSpecialty DateOfRenewal
1 97767240 Landscaping 2020-11-28

Query executed successfully. | acadsql17.asurite.ad.asu.edu | SUAD\itsanche (65) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 1 rows
```

Certificate#	TradeSpecialty	DateOfRenewal	
1	97767240	Landscaping	2020-11-28

4. Deleting an installer/technician and all their certifications with single DELETE statement - IS, VS

The screenshot shows a SQL Server Management Studio (SSMS) window. The title bar indicates the connection is to 'ACADSQL17.SP20...dbo.WORK_ORDERS'. There are two tabs open: 'Milestone4.sql' and 'Milestone3.sql'. The main pane displays the following T-SQL script:

```
1 /* Basic Test #4
2 Deleting an installer/technician and all of their certifications.*/
3 /* Inserting on delete cascade constraints to various attributes to enable the deletion
4 of an employee easily and smoothly from the database */
5 ALTER TABLE TECHNICIANS
6     DROP CONSTRAINT [FK_TECHNICIA_EmpID_671F4F74];
7 ALTER TABLE TECHNICIANS
8     ADD FOREIGN KEY (EmpID)
9         REFERENCES EMPLOYEES(EmpID)
10    ON DELETE CASCADE;
11 ALTER TABLE WORK_ORDERS
12     DROP CONSTRAINT [FK_WORK_ORDE_TechI_2645B050];
13 ALTER TABLE WORK_ORDERS
14     ADD FOREIGN KEY(TechID)
15         REFERENCES TECHNICIANS(EmpID)
16    ON DELETE CASCADE;
17 ALTER TABLE PARTS_USED
18     DROP CONSTRAINT [FK_PARTS_USE_Work0_2739D489];
19 ALTER TABLE PARTS_USED
20     ADD FOREIGN KEY(WorkOrderID)
21         REFERENCES WORK_ORDERS(WorkOrderID)
22    ON DELETE CASCADE;
```

The script consists of 22 numbered lines. Lines 1 through 4 are comments explaining the purpose and context. Lines 5 through 22 are ALTER TABLE statements that drop existing foreign key constraints and add new ones with ON DELETE CASCADE specified.

The bottom pane shows the 'Messages' window with the message: 'Command(s) completed successfully.' and the status bar at the bottom indicates: 'Query executed successfully.' and 'acadsql17.asurite.ad.asu.edu | ASUAD\vsHourie (65) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows'.

BasicTest4BEFORE.s...UAD\itsanche (87)* → X BasicTest4AFTER.sq...UAD\itsanche (71))*

```
/* Basic Test #4      BEFORE:
   Deleting an installer/technician and all of their certifications */

SELECT T.EmpID AS 'Technician ID', E.EmpFName + ' ' + E.EmpLName AS 'Technician Name', T.Certificate#
  FROM TECHNICIANS T
    JOIN EMPLOYEES E
      ON T.EmpID=E.EmpID
     WHERE T.EmpID = '8149842015';
```

100 %

Results Messages

Technician ID	Technician Name	Certificate#
8149842015	Chris Evans	30432668

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\itsanche (87) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 1 rows

BasicTest4BEFORE.s...UAD\itsanche (87)* → X BasicTest4AFTER.sq...UAD\itsanche (71))*

```
/* Basic Test #4      AFTER:
   Deleting an installer/technician and all of their certifications */

DELETE
  FROM TECHNICIANS
 WHERE EmpID = '8149842015';

SELECT T.EmpID AS 'Technician ID', E.EmpFName + ' ' + E.EmpLName AS 'Technician Name', T.Certificate#
  FROM TECHNICIANS T
    JOIN EMPLOYEES E
      ON T.EmpID=E.EmpID
     WHERE T.EmpID = '8149842015';
```

100 %

Results Messages

Technician ID	Technician Name	Certificate#
---------------	-----------------	--------------

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\itsanche (71) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows

5. For a given work order, displaying all of the parts/labor costs pertaining to it - IS

The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'FAILTestQuery10.sql - UAD\itsanche (96)*', 'BasicTest5.sql - ac...SUAD\itsanche (94)*', and 'BasicTest4BEFORE.sql - UAD\itsanche (87)'. The main area contains a query window with the following code:

```

/* Basic Test #5
For a given work order, displaying all of the parts and labor costs that pertain to it */

SELECT WO.WorkOrderID, PU.PartID, P.PartDescription AS 'Part Description', WO.TotalHours AS 'Hours Worked', P.PartPrice AS 'Part Price'
FROM WORK_ORDERS WO
JOIN PARTS_USED PU
ON PU.WorkOrderID=WO.WorkOrderID
JOIN PARTS P
ON P.PartID=PU.PartID
WHERE WO.WorkOrderID = '0245421460';

```

The results pane shows a table with the following data:

	WorkOrderID	PartID	Part Description	Hours Worked	Part Price
1	0245421460	0024568121	Hour of Labor	19.00	290.00
2	0245421460	0818355604	Air Filter 12x24x2	19.00	238.30
3	0245421460	0866242420	Air Filter 12x24x1	19.00	858.37

At the bottom, a message bar indicates: 'Query executed successfully.' and 'acadsq17.asurite.ad.asu.edu | ASUAD\itsanche (94) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 3 rows'.

6. (FAIL) Trying to delete a Customer who has past work orders - NT

The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery4.sql - ac...UAD\nvankay2 (71)*', 'SQLQuery2.sql - ac...UAD\nvankay2 (59)*', and 'SQLQuery1.sql - ac...UAD\nvankay2 (67)*'. The main area contains a query window with the following code:

```

/*6. Trying to delete a Customer who has past work orders should fail.*/
select workorderid, residentid, propertyid
from WORK_ORDERS
where residentid = '3780896439';

```

The results pane shows a table with the following data:

	workorderid	residentid	propertyid
1	5605315456	3780896439	4399332669

At the bottom, a message bar indicates: 'Query executed successfully.' and 'acadsq17.asurite.ad.asu.edu | ASUAD\nvankay2 (59) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 1 rows'.

SQLQuery3.sql - ac...UAD\nvankay2 (68)* SQLQuery2.sql - ac...UAD\ nvankay2 (59)* SQLQuery1.sql - ac...UAD\ nvankay2 (67)*

```
/*6. Trying to delete a Customer who has past work orders should fail.*/
delete
from RESIDENTS
where ResidentID = '3780896439';
```

150 %

Messages

```
Msg 547, Level 16, State 0, Line 2
The DELETE statement conflicted with the REFERENCE constraint "FK__WORK_ORDER__WorkO__6D0D32F4".
The statement has been terminated.
```

150 %

Query completed with errors. | acadsq17.asurite.ad.asu.edu | ASUAD\ nvankay2 (67) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows

7. (FAIL) Showing every employee must have an emergency contact - NT

SQLQuery7.sql - ac...UAD\ nvankay2 (55) SQLQuery6.sql - ac...UAD\ nvankay2 (52)* SQLQuery5.sql - ac...UAD\ nvankay2 (74)*

```
/*BasicTest #7*/
/*Every employee must have an emergency contact.Adding an employee without emg contact info should show an error.*/
Insert into EMPLOYEES
values ('1212447095', 'Jack', 'Bosco', '12345 North 69th Avenue, Glendale, AZ 85308', '623-456-9087', 'T', 'Donnie Bosco',
'Father', null);
Insert into EMPLOYEES
values ('8190360250', 'John', 'Mayer', '18886 North 59th Avenue, Glendale, AZ 85308', '623-111-0087', 'T', null,
'Wife', '900-156-1000');
```

100 %

Messages

```
Msg 515, Level 16, State 2, Line 3
Cannot insert the value NULL into column 'EmgContactPhone', table 'SP20_CIS365_19668_TEAM5.dbo.EMPLOYEES'; column does not allow nulls.
The statement has been terminated.
Msg 515, Level 16, State 2, Line 6
Cannot insert the value NULL into column 'EmgContactName', table 'SP20_CIS365_19668_TEAM5.dbo.EMPLOYEES'; column does not allow nulls.
The statement has been terminated.
```

100 %

Query completed with errors. | acadsq17.asurite.ad.asu.edu | ASUAD\ nvankay2 (52) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows

8. (FAIL) Entering a user into the database with a date retired before date set up will fail - VS

```

ACADSQL17_SP20_C_EAM5 - dbo.USERS      Milestone4.sql - ac...SUAD\vsjourie (77)*  X  Milestone3.sql - ac...SUAD\vsjourie (76)*

1  /* Basic Test #8
2   Every user entered into the database must have a date retired after the date the credentials were
3   set up. Entering a user into the database with a date retired before date set up should fail. */
4  Insert into USERS
5    values ('0123456788', '1212477095', 'Password', '2019-03-11', '2018-03-11');
6  Insert into USERS
7    values ('6591784581', '8190360250', 'FavCerealCheerios!', '2019-03-12', '2018-03-12');
8  Insert into USERS
9    values ('3056923968', '4808236737', 'XB2Ba8BvxL9S8Q', '2019-02-01', '2018-03-09');

140 %  ▾
Messages
Msg 547, Level 16, State 0, Line 4
The INSERT statement conflicted with the CHECK constraint "Date_Check". The conflict occurred in database "SP20_CIS365".
The statement has been terminated.
Msg 547, Level 16, State 0, Line 6
The INSERT statement conflicted with the CHECK constraint "Date_Check". The conflict occurred in database "SP20_CIS365".
The statement has been terminated.
Msg 547, Level 16, State 0, Line 8
The INSERT statement conflicted with the CHECK constraint "Date_Check". The conflict occurred in database "SP20_CIS365".
The statement has been terminated.

140 %  ▾
Query completed with errors.  acadsq17.asurite.ad.asu.edu | ASUAD\vsjourie (77) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows

```

9. (FAIL) Entering a part which does not exist in a work order will fail. - VS

```

ACADSQL17_SP20_C_EAM5 - dbo.USERS      Milestone4.sql - ac...SUAD\vsjourie (77)*  X  Milestone3.sql - ac...SUAD\vsjourie (76)*

10 /* Basic Test #9
11 Every part used in a work order must be some part which already exists in the database.
12 Using a part which does not exist in a work order will fail. */
13 Insert into PARTS_USED
14   values ('0245421460', '0866242800');
15 Insert into PARTS_USED
16   values ('7799342414', '0866240420');
17 Insert into PARTS_USED
18   values ('5605315456', '0547436819');

140 %  ▾
Messages
Msg 547, Level 16, State 0, Line 14
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__PARTS_USE__PartI__14270015". The conflict occurred
The statement has been terminated.
Msg 547, Level 16, State 0, Line 16
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__PARTS_USE__PartI__14270015". The conflict occurred
The statement has been terminated.
Msg 547, Level 16, State 0, Line 18
The INSERT statement conflicted with the FOREIGN KEY constraint "FK__PARTS_USE__PartI__14270015". The conflict occurred
The statement has been terminated.

140 %  ▾
Query completed with errors.  acadsq17.asurite.ad.asu.edu | ASUAD\vsjourie (77) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows

```

10. (FAIL) Deleting a property from PROPERTY will fail as ASSETS and WORK_ORDERS tables require PropertyID. Deleting a supplier from SUPPLIERS will fail as ASSETS_SUPPLIER and PARTS_PURCHASED tables require SupplierID. – IS

The screenshot shows a SQL query window and its corresponding output window.

Query Window (Top):

```
/* Basic Test #10
Attempting to delete a property from PROPERTY table and a supplier from SUPPLIERS table.
Deleting a property from PROPERTY will fail as ASSETS and WORK_ORDERS tables require PropertyID.
Deleting a supplier from SUPPLIERS will fail as ASSETS_SUPPLIER and PARTS_PURCHASED tables require SupplierID */

DELETE
FROM PROPERTY
WHERE PropertyID = '2556275703';

DELETE
FROM SUPPLIERS
WHERE SupplierID = '4085239879';
```

Output Window (Bottom):

Messages

```
Msg 547, Level 16, State 0, Line 7
The DELETE statement conflicted with the REFERENCE constraint "FK__ASSETS__Property__06CD04F7". The conflict occurred in database "SP20_CIS365".
The statement has been terminated.

Msg 547, Level 16, State 0, Line 11
The DELETE statement conflicted with the REFERENCE constraint "FK__PARTS_PUR__Suppl__1DB06A4F". The conflict occurred in database "SP20_CIS365".
The statement has been terminated.
```

100 %

100 %

Query completed with errors.

acadsql17.asurite.ad.asu.edu | ASUAD\itsanche (96) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows

Advanced Queries

1. GROUP BY/HAVING: List the number of incidences of each accident type with more than one accident. - JL

The screenshot shows a SQL Server Management Studio window. The top pane displays a T-SQL query:

```
SQLQuery1.sql - ac...SUAD\jalewi26 (73) 150 %
/*
Advanced Query #1
List the number of incidences of each accident type with more than one accident*
SELECT AccidentType, COUNT(AccidentType) AS "Number of Accidents"
FROM ACCIDENTS
GROUP BY AccidentType
HAVING COUNT(AccidentType) > 1
ORDER BY COUNT(AccidentType) DESC;
```

The bottom pane shows the results of the query:

AccidentType	Number of Accidents
Car crash	10
Fall from ladder	2
Fell down stairs	2

At the bottom of the window, a status bar indicates: "Query executed successfully." and "150 % acadsq17.asurite.ad.asu.edu | SUAD\jalewi26 (73) | SP20_CIS365_19668_TEAMS | 00:00:00 | 3 rows".

2. GROUP BY/HAVING: List the part ID, supplier ID, supplier name, quantity in stock, and reorder point of all parts over \$300 - JL

The screenshot shows a SQL Server Management Studio window. The top pane displays a T-SQL query:

```
SQLQuery1.sql - ac...SUAD\jalewi26 (73) 150 %
/*
Advanced Query #2
List the part ID, supplier ID, supplier name, quantity in stock, and reorder point of all parts over $300/
SELECT P.PartID, S.SupplierID, SupplierName, QuantityInStock, ReorderPoint
FROM PARTS P JOIN PARTS_PURCHASED PP
ON P.PartID = PP.PartID
JOIN SUPPLIERS S
ON PP.SupplierID = S.SupplierID
GROUP BY P.PartID, S.SupplierID, SupplierName, QuantityInStock, ReorderPoint, PartPrice
HAVING PartPrice > 300;
```

The bottom pane shows the results of the query:

PartID	SupplierID	SupplierName	QuantityInStock	ReorderPoint
1	0375200087	Robert Hubbard	442	78
2	0375200087	Amber Young	442	78
3	0547436820	John Rollins	104	38
4	0547436820	Carmie Ramirez	104	38
5	0866242420	Peter Smith	10	80
6	0866242420	Laura Stein	10	80
7	1204535835	David Cordova	611	74
8	1204535835	Richard Shelton	611	74
9	1356860564	Robert Wells	753	86
10	1356860564	Gary Jacobs	753	86
11	1665485535	Robert Wells	329	32
12	1665485535	Gary Jacobs	329	32
13	2025818582	Thomas Phillips	723	15
14	2025818582	Natalia Caldwell	723	15
15	2477146513	Daniel Watson	807	76
16	2477146513	Kathleen Kline	807	76
17	4453862646	Tanya Gay	312	82

At the bottom of the window, a status bar indicates: "Query executed successfully." and "150 % acadsq17.asurite.ad.asu.edu | SUAD\jalewi26 (73) | SP20_CIS365_19668_TEAMS | 00:00:00 | 38 rows".

3. GROUP BY/HAVING: List the work order ID, employee ID, and employee name of work orders with more than 10 total hours - JL

The screenshot shows the SQL Server Management Studio interface. In the top pane, there are three tabs: 'SQLQuery3.sql - ac...SUAD\jalewi26 (54)*' (selected), 'SQLQuery2.sql - ac...SUAD\jalewi26 (55)*', and 'SQLQuery1.sql - ac...SUAD\jalewi26 (52)*'. The 'SQLQuery3.sql' tab contains the following T-SQL code:

```

/*
Advanced Query #3
List the work order ID, employee ID, and employee name of work orders with more than 10 total hours/
*/
SELECT WorkOrderID, TechID, EmpFName, EmpLName
FROM EMPLOYEES E JOIN TECHNICIANS T
ON E.EmpID = T.EmpID
JOIN WORK_ORDERS W
ON T.EmpID = W.TechID
GROUP BY WorkOrderID, TechID, EmpFName, EmpLName, TotalHours
HAVING TotalHours > 10;

```

In the bottom pane, the 'Results' tab is selected, displaying the query results in a table:

	WorkOrderID	TechID	EmpFName	EmpLName
1	0245421460	0533456789	Dwayne	Wade
2	1642972264	8830282968	Omar	Epps
3	2393881879	0433456789	Bradley	Beal
4	4329033923	2612804505	John	Sandusky
5	5516291197	0233456789	LeBron	James
6	7174747431	8563223391	Bill	Clinton
7	8024978966	4270591322	Quentin	Tarantino
8	9101402602	3781003956	Jimmy	Butler

A status bar at the bottom indicates: 'Query executed successfully.' and 'acadsq117.asurite.ad.asu.edu... | SUAD\jalewi26 (54) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 8 rows'.

4. Nested: List first and last name(s) of technicians who have worked on a work order for more than 10 - NT

The screenshot shows the SQL Server Management Studio interface. In the top pane, there are three tabs: 'SQLQuery1.sql - ac...UAD\nvankay2 (64)*' (selected), 'SQLQuery2.sql - ac...UAD\ nvankay2 (65)*', and 'SQLQuery3.sql - ac...UAD\ nvankay2 (66)*'. The 'SQLQuery1.sql' tab contains the following T-SQL code:

```

/*
Advanced Query #4*/
/*List the first and last name(s) of technicians who have worked on a work order for more than 10 hours*/
Select empfname, emplname
from employees
where empid in
(select empid
from technicians
where empid in
(select techid
from WORK_ORDERS
where TotalHours >10));

```

In the bottom pane, the 'Results' tab is selected, displaying the query results in a table:

	empfname	emplname
1	LeBron	James
2	Bradley	Beal
3	Dwayne	Wade
4	John	Sandusky
5	Jimmy	Butler
6	Quentin	Tarantino
7	Bill	Clinton
8	Omar	Epps

A status bar at the bottom indicates: 'Query executed successfully.' and 'acadsq117.asurite.ad.asu.edu... | SUAD\ nvankay2 (64) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 8 rows'.

5. Nested: List the names of suppliers who supplied to assets located in Phoenix - NT

SQLQuery1.sql - ac...UAD\nvankay2 (64)* → X

```

/*Advanced Query #5*/
/*List the names of suppliers who supplied to assets located in phoenix*/
select suppliername 'Suppliers in Phoenix'
from SUPPLIERS
where supplierid in
(select supplierid
from asset_supplier
where assetid in
(select assetid
from ASSETS
where propertyid in
(select propertyid
from PROPERTY
where PropertyCity = 'Phoenix'))))
```

100 %

	Results	Messages
	Suppliers in Phoenix	
1	David Cordova	
2	Richard Shelton	
3	Robert Wells	
4	Daniel Watson	
5	Carmen Adams	
6	Daniel Larson	
7	Barbara Roberson	
8	Kathleen Kline	

Query executed successfully. | acadsq17.asurite.ad.asu.edu | ASUAD\ nvankay2 (64) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 8 rows

6. Nested: List all the parts used for APT#126 - NT

SQLQuery1.sql - ac...UAD\nvankay2 (69)* → X

```

/*Advanced Query #6*/
/*List all the parts used for apt#126*/
select partdescription
from PARTS
where PartID in
(select partid
from PARTS_USED
where WorkOrderID in
(select workorderid
from WORK_ORDERS
where ResidentID =
(select ResidentID
from residents
where ResApt ='126')));
```

100 %

	Results	Messages
	partdescription	
1	Air Filter 26x30x5	
2	Air Filter 30x40x10	

Query executed successfully. | acadsq17.asurite.ad.asu.edu | ASUAD\ nvankay2 (69) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 2 rows

7. VIEWS: - VS & MB

Milestone4.sql - ac...UAD\vshourie (146)* | IC09.sql - acadsq11...UAD\vshourie (160)

```

19     values ('5605315456', '0547436819');

20
21 /* Advanced Test #7 - Views (Taken from the "RESULTS" section of Milestone 2)
22 "A listing of all users and the number of days the account was active." */
23 CREATE VIEW UserAccountActivity AS
24     (SELECT UserID AS "User ID",
25      DATEDIFF(DAY,U.DateSetUpStamp,GETDATE()) AS "Number of Days Account was/is Active"
26      FROM USERS U);
27 SELECT *
28   FROM UserAccountActivity;
29
30 /* Advanced Test #9 - Views (Taken from the "RESULTS" section of Milestone 2)

```

100 %

Results

User ID	Number of Days Account was/is Active
1 0123456789	388
2 0341388010	3490
3 0765940686	2712
4 1426569593	50
5 1441032341	2200
6 1482310915	3088
7 1640700292	1912
8 1845314698	4042
9 2777640988	92
10 3007276158	487
11 3056923969	426

Query executed successfully. | acadsq11.asurite.ad.asu.edu | ASUAD\vshourie (146) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 30 rows

8. VIEWS: MB

SQLQuery1.sql - a...UAD\mbrewer7 (97)*

```

/*Advanced Query #8
Total number of accidents per month for the past year by accident status. */
create view MonthsWithAccidents as
    select DATENAME(month, OccurrenceDate) as "Months with Accidents",
           COUNT(DATENAME(month, OccurrenceDate)) as "Number of Accidents",
           Status
      from ACCIDENTS
     where OccurrenceDate > DATEADD(year,-1,GETDATE())
       group by DATENAME(month, OccurrenceDate), Status;

select *
  from MonthsWithAccidents
 order by Status;

```

100 %

Results

Months with Accidents	Number of Accidents	Status
1 December	1	Emergency
2 June	1	Minor
3 October	1	Minor

9. VIEWS: VS & MB

The screenshot shows a SQL Server Management Studio (SSMS) window. At the top, there are two tabs: 'Milestone4.sql - ac...UAD\vsHourie (146)*' and 'IC09.sql - acadsq1...UAD\vsHourie (160)'. The code editor contains the following SQL script:

```
29  /* Advanced Test #9 - Views (Taken from the "RESULTS" section of Milestone 2)
30  "Residents who have had maintenance work completed in the past 6 months." */
31  CREATE VIEW ResidentsRecentMaintenance AS
32      (SELECT R.ResidentID AS "Resident ID",
33          R.ResApt AS "Resident's Apartment #",
34          R.ResName AS "Resident's Name",
35          W.WorkOrderID AS "Work Order ID",
36          W.LastMaintDate AS "Date Last Maintenance Occurred"
37      FROM RESIDENTS R JOIN WORK_ORDERS W
38          ON R.ResidentID = W.ResidentID
39          WHERE DATEDIFF(MONTH,W.LastMaintDate,GETDATE()) <= 6);
40  SELECT *
41  FROM ResidentsRecentMaintenance;
```

The results pane below the code editor displays a table with the following data:

	Resident ID	Resident's Apartment #	Resident's Name	Work Order ID	Date Last Maintenance Occurred
1	3673914534	410	Sophia Jackson	0245421460	2019-12-06
2	2263909583	316	Ruth Greene	1858869730	2020-08-22
3	9003459567	834	Zachary Marsh	2393881879	2020-04-07
4	5252053437	628	Brandon Pena	7174747431	2020-12-06
5	2640852291	766	Jennifer Smith	7491160348	2020-02-05
6	3498751707	286	Kristina Rangel	8893160795	2020-01-06

At the bottom of the results pane, there is a status bar with the following information: Query... | acadsq17.asurite.ad.asu.edu | ASUAD\vsHourie (146) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 6 rows

Transactions

/* Varun Shourie */

M5SqlQueries.sql ...UAD\vshtourie (141)* # X

```

1 /* 1. When a user ID is retired, an employee must be deleted or flagged as inactive. If deleted, this requires an on delete cascade
2 constraint with the ACCIDENTS table, which is likely undesirable for the client. If flagged as inactive, the EMPLOYEES table must
3 be altered to include an attribute which flags the status of the employee. The following code adds the flag attribute to the EMPLOYEES
4 table, making question number one possible. */
5 ALTER TABLE EMPLOYEES           -- Adds the flag attribute to the EMPLOYEES table.
6   ADD DatabaseAccessStatus varchar(8);
7 UPDATE EMPLOYEES               -- Updates all employees' statuses with active access to the database.
8   SET DatabaseAccessStatus = 'Active'
9   WHERE EmpID IN
10    (SELECT u.EmpID
11     FROM USERS u
12      WHERE DateRetired IS NULL);
13 UPDATE EMPLOYEES               -- Updates all employees' statuses who once had access to the database.
14   SET DatabaseAccessStatus = 'Inactive'
15   WHERE EmpID IN
16    (SELECT u.EmpID
17     FROM USERS u
18      WHERE DateRetired IS NOT NULL);
19 UPDATE EMPLOYEES               -- Updates all employees' statuses who never had access to the database.
20   SET DatabaseAccessStatus = 'N/A'
21   WHERE EmpID IN
22    (SELECT e.EmpID
23     FROM EMPLOYEES e LEFT JOIN USERS u
24       ON e.EmpID = u.EmpID
25         WHERE u.UserID IS NULL);
26 SELECT * FROM EMPLOYEES;

```

100 %

Results Messages

EmpFName	EmpLName	EmpAddress	EmpPhone#	EmpJobType	EmgContactName	EmgRelationship	EmgContactPhone	DatabaseAccessStatus
Anna	Garcia	10024 South Fountain Avenue, Pasadena, CA, 81003	230-743-8235	O	Andre Garcia	Husband	155-520-7776	N/A
John	Smith	1234 East Adams Street, Phoenix, AZ 99921	333-345-5555	T	Jane Smith	Mother	222-222-2222	N/A

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\vshtourie (141) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 61 rows

M5SqlQueries.sql ...SUAD\vshtourie (61)* # X

```

28 /* 1. When an employee has left the company, retire their userID and password so that they can no longer
29 access company information and delete or flag the employee as inactive. BEFORE Screenshot */
30 SELECT *
31   FROM EMPLOYEES
32   WHERE EmpID = '5425516460';
33 SELECT *
34   FROM USERS
35   WHERE EmpID = '5425516460';
36 /* 1. When an employee has left the company, retire their userID and password so that they can no longer

```

100 %

Results Messages

EmpID	EmpFName	EmpLName	EmpAddress	EmpPhone#	EmpJobType	EmgContactName	EmgRelationship	EmgContactPhone	DatabaseAccessStatus
5425516460	Oscar	Mayer	97212 West Paradise Lane, Glendale, AZ 85308	925-449-4190	T	Amelia Mayer	Mother	847-390-5539	Active

UserID	EmpID	UserPassword	DateSetUpStamp	DateRetired
0341388010	5425516460	lr4rbBE+0_A7WL	2010-09-12	NULL

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\vshtourie (61) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 2 rows

M5SqlQueries.sql -...SUAD\vshourie (61)* ➔ X

```
36 /* 1. When an employee has left the company, retire their userID and password so that they can no longer
37 access company information and delete or flag the employee as inactive. AFTER Screenshot */
38 BEGIN TRANSACTION;
39 UPDATE EMPLOYEES
40     SET DatabaseAccessStatus = 'Inactive'
41     WHERE EmpID = '5425516460';
42 UPDATE USERS
43     SET DateRetired = GETDATE()
44     WHERE EmpID = '5425516460';
45 SELECT *
46     FROM EMPLOYEES
47     WHERE EmpID = '5425516460';
48 SELECT *
49     FROM USERS
50     WHERE EmpID = '5425516460';
51 ROLLBACK; -- If there is an error and we are not satisfied, we should choose this option.
52 COMMIT; -- If we are satisfied with the data, we should choose this option.
```

100 %

Results Messages

Address	EmpPhone#	EmpJobType	EmgContactName	EmgRelationship	EmgContactPhone	DatabaseAccessStatus
Vest Paradise Lane, Glendale, AZ 85308	925-449-4190	T	Amelia Mayer	Mother	847-390-5539	Inactive

	UserID	EmpID	UserPassword	DateSetUpStamp	DateRetired
1	0341388010	5425516460	lr4rbBEj+0_A7WL	2010-09-12	2020-04-22

Query executed successfully. | acadsq17.asurite.ad.asu.edu | ASUAD\vshourie (61) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 2 rows

/* Mat Brewer */

SQLQuery1.sql - a...UAD\mbrewer7 (82)* ➔ X

```

/* Section 1. (10) Transactions Overview:
Recall that transactions include several SQL statements coded to run in a batch execution – all statements in
the batch execute successfully (commit), or none should be executed (rollback). Your team will create 2
separate transactions using the tables/views in your project database and display the before/after screenshots
to show that the transaction was successful.

2. A new supplier is added to the database. Include all parts (at least 3) that can be supplied by the new
supplier. */

--BEFORE
select *
from SUPPLIERS
where SupplierID = '1234567890';

select *
from PARTS
where PartID = '1234567891'
or PartID = '1234567892'
or PartID = '1234567893';

select *
from PARTS_PURCHASED
where PartID = '1234567891'
or PartID = '1234567892'
or PartID = '1234567893';

```

100 %

SupplierID	SupplierName	SupplierAddress	SupplierCity	SupplierState	SupplierZip	SupplierPhone	SupplierContact	WebAddress	DeliveryPriority

PartID	PartCategory	PartSubCategory	PartDescription	QuantityInStock	SpecialFeatures	ReorderPoint	ReorderQuantity	PartPrice

PartID	SupplierID

```

SQLQuery1.sql - a...UAD\mbrewer7 (82)*  ✎ X

BEGIN TRANSACTION;

insert into SUPPLIERS values ('1234567890', 'Daniel Jones', '11198 East Camina Plata', 'Queen Creek', 'AZ', '85142', '480-924-7013', 'Rock Lee', 'danieljones@gmail.com', 'on-time');

insert into PARTS values('1234567891', 'Pumps', '1" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', '5', NULL, '2', '2', 450.00);
insert into PARTS values('1234567892', 'Pumps', '1.5" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', '5', NULL, '2', '2', 500.00);
insert into PARTS values('1234567893', 'Pumps', '3" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', '5', NULL, '3', '3', 550.00);

insert into PARTS_PURCHASED values('1234567891', '1234567890');
insert into PARTS_PURCHASED values('1234567892', '1234567890');
insert into PARTS_PURCHASED values('1234567893', '1234567890');

--AFTER
select *
from SUPPLIERS
where SupplierID = '1234567890';
select *
from PARTS
where PartID = '1234567891'
or PartID = '1234567892'
or PartID = '1234567893';
select *
from PARTS_PURCHASED
where PartID = '1234567891'
or PartID = '1234567892'
or PartID = '1234567893';

ROLLBACK; --in case there is an error
COMMIT;

```

100 %

	SupplierID	SupplierName	SupplierAddress	SupplierCity	SupplierState	SupplierZip	SupplierPhone	SupplierContact	WebAddress	DeliveryRating
1	1234567890	Daniel Jones	11198 East Camina Plata	Queen Creek	AZ	85142	480-924-7013	Rock Lee	danieljones@gmail.com	on-time

	PartID	PartCategory	PartSubCategory	PartDescription	QuantityInStock	SpecialFeatures	ReorderPoint	ReorderQuantity	PartPrice
1	1234567891	Pumps	1" Diaphragm Pump	Stainless Steel w/ Santoprene Diaphragm	5	NULL	2	2	450.00
2	1234567892	Pumps	1.5" Diaphragm Pump	Stainless Steel w/ Santoprene Diaphragm	5	NULL	2	2	500.00
3	1234567893	Pumps	3" Diaphragm Pump	Stainless Steel w/ Santoprene Diaphragm	5	NULL	3	3	550.00

	PartID	SupplierID
1	1234567891	1234567890
2	1234567892	1234567890
3	1234567893	1234567890

Trigger

/* Varun Shourie, Isis Sanchez */

```

SQLQuery1.sql - ac...UAD\vshourie (88) M5SqlQueries.sql -...SUAD\vshourie (68)* -> X Milestone3.sql - ac...SUAD\vshourie (65)*
135 /* 1. (10) TRIGGER: When a new work order is created (inserted), automatically assign the first available technician to do the job; assign
136 | a tech with the appropriate tech specialty (if the task is plumbing, don't assign an electrician). */
137 /* Before we create the trigger, we must create a way to flag which technicians are available and which are unavailable from working
138 | on a work order. To do this, we must create a flag attribute as we did in the first question. */
139 ALTER TABLE TECHNICIANS          -- Adds the flag attribute to the TECHNICIANS table
140   ADD AvailabilityStatus varchar(11); -- We assume a technician is available when they are added.
141 UPDATE TECHNICIANS             -- Since all work orders are finished in the Work Orders Table,
142   SET AvailabilityStatus = 'Available'    -- all technicians are available.
143   WHERE EmpID IN
144     (SELECT TechID
145      FROM WORK_ORDERS
146      WHERE EndDate IS NOT NULL);
147
148 UPDATE TECHNICIANS           -- Since all work orders are finished in the Work Orders Table,
149   SET AvailabilityStatus = 'Unavailable' -- no technicians should be unavailable.
150   WHERE EmpID IN
151     (SELECT TechID
152      FROM WORK_ORDERS
153      WHERE EndDate IS NOT NULL);
154
155 SELECT * FROM TECHNICIANS;

```

Results

EmpID	TradeSpecialty	Certificate#	DateOfRenewal	DateOfOrigCertificate	AvailabilityStatus
1 0123456789	HVAC	77821236	2021-10-10	2019-10-10	Available
2 0133456789	Pools	75288777	2022-01-23	2000-05-15	Available
3 0233456789	Landscaping	97767240	2020-11-28	2008-11-28	Available
4 0333456789	Carpentry	38336214	2020-05-28	2015-05-28	Available
5 0433456789	Landscaping	49066751	2020-05-16	2006-05-16	Available

Query executed successfully.

```

SQLQuery1.sql - ac...UAD\vshourie (88) M5SqlQueries.sql -...SUAD\vshourie (68)* -> X Milestone3.sql - ac...SUAD\vshourie (65)*
154
155 SELECT * FROM TECHNICIANS;
156
157 /* Next, we must insert a value to into Tasks which shows the correct Task Type. If a correct task type is provided, it can be used to
158 | match an available technician. */
159 INSERT INTO TASKS
160   VALUES('1039259302','HVAC','Repair Damaged Air Filters', 'Wear and Tear');
161
162 SELECT * FROM TASKS;

```

Results

TaskID	TaskType	TaskDescription	RootCause
1 1039259302	HVAC	Repair Damaged Air Filters	Wear and Tear
2 1239259302	C06	Restroom Pm. Route	NULL
3 1611618056	Pool - Weekly	Swimming Pool	NULL
4 2371053656	Yearly Building Inspection	Building 102	NULL
5 2463146448	C04	Cafeteria Water Filters	NULL
6 2665931830	Fire Hose 1 Year Insp...	Fire Hose Cabinet, Bldg ...	NULL
7 2749370837	Repair	Front Parking Lot - 50 sp...	NULL
8 2764771170	Repair	Room 3	Fire damage
9 2813858535	C07	Cafeteria Ice Machine	NULL
10 2987970115	Yearly Building Inspection	Building Number 101	NULL
11 4117998742	Repair	Front Gate	car crashed i...
12 4336837906	Fire Hose 1 Year Insp...	Fire Hose Cabinet, Bldg ...	NULL
13 4423222485	Repair	Door lock jammed	Misuse
14 4424071352	Yearly Building Inspection	Building Number 103	NULL
15 4599572123	Repair	Cafeteria Refrig & Freeze...	Wear/Normal...

Query executed successfully.

SQLQuery1.sql - ac...UAD\vshourie (88) M5SqlQueries.sql -...SUAD\vshourie (68)* X Milestone3.sql - ac...SUAD\vshourie (65)*

```

198 /* Creating the trigger */
199 CREATE TRIGGER TechniciansWOInsert
200 ON WORK_ORDERS
201 AFTER INSERT
202 AS
203 UPDATE WORK_ORDERS
204 SET TechID =
205 (SELECT TOP 1 EmpID
206 FROM TECHNICIANS
207 WHERE TradeSpecialty IN
208 (SELECT TaskType
209 FROM TASKS
210 WHERE TaskID IN
211 (SELECT TaskID
212 FROM inserted)
213 AND AvailabilityStatus = 'Available'))
214 WHERE WorkOrderID IN
215 (SELECT WorkOrderID
216 FROM inserted);
217 UPDATE TECHNICIANS
218 SET AvailabilityStatus = 'Unavailable'
219 WHERE EmpID IN
220 (SELECT TechID
221 FROM WORK_ORDERS
222 WHERE WorkOrderID IN
223 (SELECT WorkOrderID
224 FROM inserted));

```

100 %

Messages

Query executed successfully.

SQLQuery1.sql - ac...UAD\vshourie (88) M5SqlQueries.sql -...SUAD\vshourie (68)* X

```

198 /* Creating the trigger (with successful execution message) */
199 CREATE TRIGGER TechniciansWOInsert
200 ON WORK_ORDERS
201 AFTER INSERT
202 AS
203 UPDATE WORK_ORDERS
204 SET TechID =
205 (SELECT TOP 1 EmpID
206 FROM TECHNICIANS
207 WHERE TradeSpecialty IN
208 (SELECT TaskType
209 FROM TASKS
210 WHERE TaskID IN
211 (SELECT TaskID
212 FROM inserted))
213 AND AvailabilityStatus = 'Available');

```

100 %

Messages

Command(s) completed successfully.

Query executed successfully.

SQLQuery1.sql - ac...UAD\vshourie (88) M5SqlQueries.sql - ...SUAD\vshourie (68)* Milestone3.sql - ac...SUAD\vshourie (65)*

```

225  /* 1. (10) TRIGGER: When a new work order is created (inserted), automatically assign the first available technician to do the job; assign
226  a tech with the appropriate tech specialty (if the task is plumbing, don't assign an electrician). */
227  /* Before screenshot */
228  SELECT *
229  FROM WORK_ORDERS
230  WHERE WorkOrderID = '8437549859';
231
232  SELECT * FROM TECHNICIANS;
233

```

Results

WorkOrderID	ResidentID	PropertyID	TaskID	TechID	CreatorID	DateCreated	DesiredWorkDate	StartDate	EndDate	LastMaintDate	Emergency	TotalHours	WorkOrderNotes
1	0123456789	HVAC	77821236	2021-10-10	2019-10-10								
2	0133456789	Pools	75288777	2022-01-23	2000-05-15								
3	0233456789	Landscaping	97767240	2020-11-28	2008-11-28								
4	0333456789	Carpentry	38336214	2020-05-28	2015-05-28								
5	0433456789	Landscaping	49066751	2020-05-16	2006-05-16								
6	0485646926	HVAC	36392361	2020-12-16	2012-12-16								
7	0490721952	HVAC	76484817	2020-10-17	2012-10-17								
8	0533456789	Carpentry	79453766	2020-08-16	2014-01-06								
9	1212477095	HVAC	34378659	2021-11-19	2013-11-19								
10	1231737523	Carpentry	70459729	2020-07-12	2012-07-12								

Query executed successfully.

M5SqlQueries.sql - ...SUAD\vshourie (68)* Milestone3.sql - ac...SUAD\vshourie (65)*

```

233
234  /* 1. (10) TRIGGER: When a new work order is created (inserted), automatically assign the first available technician to do the job;
235  assign a tech with the appropriate tech specialty (if the task is plumbing, don't assign an electrician). */
236  /* After screenshot */
237  INSERT INTO WORK_ORDERS
238  VALUES ('8437549859', '6393256990', '9534546401', '1039259302', NULL, '3663027200', '2020-04-28', '2020-04-28',
239  '2020-05-05', NULL, NULL, '0', 23.38, 'Damaged Filter', 'Completed');
240  SELECT * FROM WORK_ORDERS
241  WHERE WorkOrderID = '8437549859';
242  SELECT * FROM TECHNICIANS;

```

Results

WorkOrderID	ResidentID	PropertyID	TaskID	TechID	CreatorID	DateCreated	DesiredWorkDate	StartDate	EndDate	LastMaintDate	Emergency	TotalHours	WorkOrderNotes
1	8437549859	6393256990	9534546401	1039259302	0123456789	3663027200	2020-04-28	2020-04-28	2020-05-05	NULL	NULL	0	23.38

EmpID	TradeSpecialty	Certificate#	DateOfRenewal	DateOfOrigCertificate	AvailabilityStatus
1	0123456789	HVAC	77821236	2021-10-10	2019-10-10
2	0133456789	Pools	75288777	2022-01-23	2000-05-15
3	0233456789	Landscaping	97767240	2020-11-28	2008-11-28
4	0333456789	Carpentry	38336214	2020-05-28	2015-05-28
5	0433456789	Landscaping	49066751	2020-05-16	2006-05-16
6	0485646926	HVAC	36392361	2020-12-16	2012-12-16
7	0490721952	HVAC	76484817	2020-10-17	2012-10-17
8	0533456789	Carpentry	79453766	2020-08-16	2014-01-06
9	1212477095	HVAC	34378659	2021-11-19	2013-11-19
10	1231737523	Carpentry	70459729	2020-07-12	2012-07-12

Query executed successfully.

Stored Procedure

/* Isis Sanchez */

```
Procedure1.sql - a...SUAD\itsanche (73)* ✎ X
/* (10) PROCEDURE: When a work order status changes to 'delayed', add a record to the DelayedWOLog record WO#, StartDate, tech working on the order (ID), status change date (today's date), root cause for the WO, and reason for the delay.

PROCEDURE: Creating table DelayedWOLog to log when WorkOrder Status is changed to delayed */

CREATE TABLE DelayedWOLog (
    WO#           char(10),
    StartDate     date,
    TechID        char(10),
    StatusChangeDate datetime      DEFAULT GETDATE(),
    RootCause     varchar(1000),
    DelayReason   varchar(1000));

SELECT *
    FROM DelayedWOLog;
```

100 %

	Results	Messages
WO# StartDate TechID StatusChangeDate RootCause DelayReason		

Query executed successfully. | acadsq17.asurite.ad.asu.edu | ASUAD\itsanche (73) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 0 rows


```
Procedure1.sql - a...SUAD\itsanche (84)* ✎ X SQLQuery1.sql - ac...SUAD\itsanche (82)
/* (10) PROCEDURE: When a work order status changes to 'delayed', add a record to the DelayedWOLog record WO#, StartDate, tech working on the order (ID), status change date (today's date), root cause for the WO, and reason for the delay.

PROCEDURE: Creating procedure AddDelayedWO to update DelayedWOLog */

CREATE PROCEDURE AddDelayedWO
    @WorkOrderID    char(10),
    @DelayReason    varchar(1000)
AS
    INSERT INTO DelayedWOLog(WO#, StartDate, TechID, StatusChangeDate, RootCause, DelayReason)
        VALUES (@WorkOrderID,
                (SELECT StartDate
                    FROM WORK_ORDERS
                    WHERE WorkOrderID = @WorkOrderID
                    AND (WorkOrderStatus = 'Delayed' OR WorkOrderStatus = 'delayed')),
                (SELECT TechID
                    FROM WORK_ORDERS
                    WHERE WorkOrderID = @WorkOrderID
                    AND (WorkOrderStatus = 'Delayed' OR WorkOrderStatus = 'delayed')),
                GETDATE(),
                (SELECT RootCause
                    FROM TASKS
                    WHERE TaskID IN
                        (SELECT TaskID
                            FROM WORK_ORDERS
                            WHERE WorkOrderID = @WorkOrderID
                            AND (WorkOrderStatus = 'Delayed' OR WorkOrderStatus = 'delayed'))),
                @DelayReason);

    EXEC AddDelayedWO 2393881879, 'Improper tools';
    EXEC AddDelayedWO 9375544818, 'Missing part';
    SELECT *
        FROM DelayedWOLog
```

80 %

	Results	Messages
WO# StartDate TechID StatusChangeDate RootCause DelayReason		
1 2393881879 2020-04-06 0433456789 2020-04-26 12:01:43.417 car crashed into it Improper tools		
2 9375544818 2002-07-27 2051366864 2020-04-26 12:01:46.580 NULL Missing part		

Query executed successfully. | acadsq17.asurite.ad.asu.edu | ASUAD\itsanche (84) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 2 rows

Indexes with Query Plans

/* Mat Brewer */

The screenshot shows a SQL Server Management Studio (SSMS) interface. At the top, there's a toolbar with icons for file operations like New, Open, Save, and Print. Below the toolbar is a menu bar with options like File, Edit, View, Tools, Options, Help, and a dropdown for Database. The main area has tabs for Results, Messages, and Execution plan. The Results tab is active and displays two tables of data.

```

SQLQuery2.sql - a..UAD\mbrewer7 (63)* X
SET STATISTICS TIME ON;
/*1. (2) Identify the attributes (at least 2) that should have secondary indexes defined (cluster index is already created) to speed up the query you will define in #3. Take a snapshot of the tables you will use (
select * from each table and show the attributes and a few rows of data in each one). */
SELECT *
FROM WORK_ORDERS
SELECT *
FROM RESIDENTS
  
```

Below the results grid, there's a status bar showing "100 %".

	WorkOrderID	ResidentID	PropertyID	TaskID	TechID	CreatorID	DateCreated	DesiredWorkDate	StartDate	EndDate	LastMaintDate	Emergency	TotalHours	WorkOrderNotes	WorkOrderStatus
1	0245421460	3673914534	4974134203	7188818037	0533456789	4723257262	2019-11-01	2019-11-02	2019-12-03	2019-12-05	2019-12-06	0	19.00	Pump replacement	completed
2	1329486777	5983055224	1778307452	9599810135	0133456789	7461707115	2019-03-18	2019-04-18	2019-04-18	2019-04-18	2019-04-18	0	7.23	Bldg. 102 Inspection	completed
3	1642972264	8270971640	2556275703	6694565143	8830282968	8606515777	2018-11-14	2018-12-14	2018-12-15	2018-12-16	2018-12-16	0	17.32	Patch ceiling	completed
4	1665200201	1557000211	1475000200	7001000200	1000200200	1000200200	2000-07-20	2000-07-20	2000-07-20	2000-07-20	2000-07-20	0	0.00	L T L	L L L

	ResidentID	ResApt	ResName	ResPhone	ResMobilePhone	ResEmail
1	1594262954	523	Marcus Mitchell	275-680-2456	276-680-2456	marc_mitchell@gmail.com
2	1766091032	206	Nathan Jones	581-485-4901	341-483-9189	nathanjones@gmail.com
3	2263909583	316	Ruth Greene	315-056-4997	NULL	ruthgreene@gmail.com
4	2640852291	766	Jennifer Smith	174-340-4593	750-540-4234	jennifersmith@gmail.com
5	3308976270	444	Glenn Clarke	NULL	258-575-0306	glennclarke@gmail.com
6	3498751707	286	Kristina Rangel	956-337-2629	NULL	kristinarangel@gmail.com
7	3673914534	410	Sophia Jackson	821-934-2000	103-325-0032	sophiajackson3@gmail.com
8	3780896439	397	Kiara Johnson	327-544-3537	326-544-3537	kjohnson4@gmail.com
9	3786184210	198	Jennifer Gardner	583-911-0103	847-004-1551	iennifergardner@gmail.com

SQLQuery1.sql - a...UAD\mbrewer7 (89)* X

```
/* 2. (2) Write the SQL commands to create the secondary indexes for the 2 attributes identified in #1.  
Give reasons for each attribute selected (use comments). Take a snapshot of the SQL code for both indexes. */  
--StartDate is a constraint in the Where clause, and using an index could reduce sorting and processing time.  
CREATE INDEX StartDateIDX ON WORK_ORDERS(StartDate);  
  
--ResName is a constraint in the ORDER BY clause, indexing ResName could reduce sorting and processing time as well.  
CREATE INDEX ResNameIDX ON RESIDENTS(ResName);
```

100 %

Messages

Command(s) completed successfully.

```
/* Query run without index and optimization with nested query to show initial statistics on time and query plan */
```

SQLQuery1.sql - a...UAD\mbrewer7 (78)* ▾ X

```
/*3. (2) Create an advanced join query and execute (use any multiple tables you wish). Show the query, the data results, the time to complete, and the query plan (take snapshots of each). The idea here is to create a query that will tax the computer resources – force the DBMS to use its resources. */

SELECT r.ResidentID AS "ID", r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT
    FROM RESIDENTS r
        JOIN WORK_ORDERS w
            ON r.ResidentID = w.ResidentID
        JOIN PROPERTY p
            ON w.PropertyID = p.PropertyID
    WHERE YEAR(StartDate) = 2019
        AND YEAR(EndDate) = 2019
        AND p.PropertyState IN ('AZ')
        AND ResApt > 100
    ORDER BY ResName;
```

100 % ◀

Results Messages

ID	ResApt	ResName	ResMobilePhone	ResEmail
1	5983055224	684	Elizabeth Moore	808-016-8528 elizabethmoore@gmail.com
2	7253005310	137	Jordan Miller	893-376-6982 jordanmiller@gmail.com
3	3673914534	410	Sophia Jackson	103-325-0032 sophiajackson3@gmail.com
4	4923659143	561	Steven Lin	NULL steve.lin@gmail.com

SQLQuery2.sql - a...UAD\mbrewer7 (63)* # X

```
/*3. (2) Create an advanced join query and execute (use any multiple tables you wish). Show the query, the data results, the time to complete, and the query plan (take snapshots of each). The idea here is to create a query that will tax the computer resources - force the DBMS to use its resources. */

SELECT r.ResidentID AS "ID", r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT
FROM RESIDENTS r
JOIN WORK_ORDERS w
ON r.ResidentID = w.ResidentID
JOIN PROPERTY p
ON w.PropertyID = p.PropertyID
WHERE YEAR(StartDate) = 2019
AND YEAR(EndDate) = 2019
AND p.PropertyState IN ('AZ')
AND ResApt > 100
ORDER BY ResName;
```

100 %

Results Messages Execution plan

```
SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time:
CPU time = 9 ms, elapsed time = 9 ms.

(4 row(s) affected)
```

SQLQuery2.sql - a...UAD\mbrewer7 (63)* ✎ X

```

/*3. (2) Create an advanced join query and execute (use any multiple tables you wish). Show the query, the
data results, the time to complete, and the query plan (take snapshots of each). The idea here is to
create a query that will tax the computer resources - force the DBMS to use its resources. */

SELECT r.ResidentID AS "ID", r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT
FROM RESIDENTS r
JOIN WORK_ORDERS w
ON r.ResidentID = w.ResidentID
JOIN PROPERTY p
ON w.PropertyID = p.PropertyID
WHERE YEAR(StartDate) = 2019
AND YEAR(EndDate) = 2019
AND p.PropertyState IN ('AZ')
AND ResApt > 100
ORDER BY ResName;

```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```

SELECT r.ResidentID AS "ID", r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT FROM RESIDENTS r JOIN WORK_ORDERS w ON r.ResidentID = w.ResidentID JOIN PROPERTY

```

```

graph TD
    A[SELECT Cost: 0 $] -- Nested Loops (Inner Join) Cost: 0 $ --> B[Nested Loops (Inner Join) Cost: 2 $]
    B --> C[Filter Cost: 0 $]
    C --> D[Compute Scalar Cost: 0 $]
    D -- Nested Loops (Inner Join) Cost: 1 $ --> E[Index Scan (NonClustered) [RESIDENTS].[ResNameIDX] {r} Cost: 17 $]
    E --> F[Key Lookup (Clustered) [RESIDENTS].[PK_RESIDENT_07FB00FC... Cost: 41 $]
    F --> G[Clustered Index Scan (Clustered) [WORK_ORDERS].[PK_WORK_ORD_AE7551... Cost: 22 $]
    G --> H[Clustered Index Seek (Clustered) [PROPERTY].[PK_PROPERTY_70C9A765C... Cost: 17 $]

```

/* Query optimized with the nested query and index is run to show a potential improvement in performance */

The screenshot shows a SQL Server Management Studio (SSMS) interface. At the top, a title bar reads "SQLQuery1.sql - a...UAD\mbrewer7 (78)*". Below the title bar is a code editor containing a SQL query. The code is a nested SELECT statement. It starts with a FROM clause for RESIDENTS, followed by WHERE clauses for ResApt > 100 and ResidentID in a subquery. The subquery selects ResidentID from WORK_ORDERS where the year of StartDate is 2019 and the year of EndDate is also 2019. It then joins PROPERTY p on PropertyID and filters for PropertyState in ('AZ'). The main query then joins RESIDENT r on ResidentID and retrieves columns ResidentID, ResApt, ResName, ResMobilePhone, and ResEmail.

```
/*4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and the query plan (take snapshots of each). */
SELECT r.ResidentID, r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT
FROM RESIDENTS r
WHERE ResApt > 100
AND r.ResidentID IN
    (SELECT w.ResidentID
     FROM WORK_ORDERS w
     WHERE YEAR(StartDate) = 2019
     AND YEAR(EndDate)= 2019
     AND w.PropertyID IN
         (SELECT p.PropertyID
          FROM PROPERTY p
          WHERE p.PropertyState IN ('AZ')))
```

The results grid below the code editor displays four rows of data:

	ResidentID	ResApt	ResName	ResMobilePhone	ResEmail
1	3673914534	410	Sophia Jackson	103-325-0032	sophiajackson3@gmail.com
2	4923659143	561	Steven Lin	NULL	steve.lin@gmail.com
3	5983055224	684	Elizabeth Moore	808-016-8528	elizabethmoore@gmail.com
4	7253005310	137	Jordan Miller	893-376-6982	jordanmiller@gmail.com

SQLQuery2.sql - a...UAD\mbrewer7 (63)* X

```
/*4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and the query plan (take snapshots of each). */
SELECT r.ResidentID, r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT
FROM RESIDENTS r
WHERE ResApt > 100
AND r.ResidentID IN
(SELECT w.ResidentID
FROM WORK_ORDERS w
WHERE YEAR(StartDate) = 2019
AND YEAR(EndDate)= 2019
AND w.PropertyID IN
(SELECT p.PropertyID
FROM PROPERTY p
WHERE p.PropertyState IN ('AZ'))))
ORDER BY ResName
```

100 % ▾

Results Messages Execution plan

```
SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time:
CPU time = 16 ms, elapsed time = 18 ms.

(4 row(s) affected)
```

SQLQuery2.sql - a...UAD\mbrewer7 (63)*

```

/*4. (2) Optimize & execute the query in #3. Show the query, the data results, the time to complete, and
the query plan (take snapshots of each). */
SELECT r.ResidentID, r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT
FROM RESIDENTS r
WHERE ResApt > 100
AND r.ResidentID IN
(SELECT w.ResidentID
FROM WORK_ORDERS w
WHERE YEAR(StartDate) = 2019
AND YEAR(EndDate)= 2019
AND w.PropertyID IN
(SELECT p.PropertyID
FROM PROPERTY p
WHERE p.PropertyState IN ('AZ')))

ORDER BY ResName

```

100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```

SELECT r.ResidentID, r.ResApt, r.ResName, r.ResMobilePhone, r.ResEmail --RESIDENT FROM RESIDENTS r WHERE ResApt > 100 AND r.ResidentID IN (

```

```

graph TD
    A[SELECT Cost: 0 %] --> B[Nested Loops (Left Semi Join) Cost: 0 %]
    B --> C[Filter Cost: 0 %]
    C --> D[Compute Scalar Cost: 0 %]
    D --> E[Nested Loops (Inner Join) Cost: 1 %]
    E --> F[Index Scan (NonClustered) [RESIDENTS].[ResNameIDX] [r] Cost: 16 %]
    F --> G[Key Lookup (Clustered) [RESIDENTS].[PK_RESIDENT_07FB00FC... Cost: 38 %]
    G --> H[Nested Loops (Inner Join) Cost: 2 %]
    H --> I[Clustered Index Scan (Clustered) [WORK_ORDERS].[PK_WORK_ORD_AE7551... Cost: 20 %]
    I --> J[Clustered Index Seek (Clustered) [PROPERTY].[PK_PROPERTY_70C9A755C... Cost: 22 %]

```

/* Varun Shourie

5. (2) Comment on the differences between the results in parts 3 & 4. How did you expect the query plans to be different? Did you gain time, or lose time – why? Based on the query that you executed in part 3, what did you expect to happen and why? */

In part 3, we observed that the SQL server parse and compile term was a time of 9ms, elapsed. In part 4, the time doubled with a parse and compile time of approximately 18ms elapsed.

Furthermore, our query plans were significantly different. In part 3, we observed that a non-clustered index scan (17%) and a clustered key lookup (41%) happened simultaneously in our query execution, taking up most of the computing resources. Next, two clustered index scans (22% and 17%) occurred at different stages with inner joins, a compute scalar, and a filter in between taking up minimal processing resources. In contrast, part 4 entails a greater amount of processing occurring at once. This is evidenced by how the non clustered index scan (16%) and the clustered key lookup (38%) occur at once along with both other clustered index scans (20% and 22%) occurring at once.

As shown by the query plan, we likely gained time in part 4 because much of the processing power was devoted to tasks being done parallel to each other rather than in sequence. Furthermore, our data set was too small (mostly tables with 30 tuples) and containing repeating values; Dr. Moser's recommended guidelines suggest indexes should be used in tables with more than a hundred tuples where values are very unique and hard to find. Today's computers function a lot more powerfully in comparison to computers of the previous century; therefore, the DBMS with today's computing resources may function more efficiently with joins than nested queries/indexes in many situations.

Based on the query plan from part 3, we expected the parse and compile plan to reduce slightly by 2-3 ms and the query plan to lengthen out slightly. This is because, as per our previous examinations, nested queries tend to distribute the resources of processing power, making it easier on the DBMS. Also, the amount of reads decrease since the DBMS does not perform cartesian products of the tables.

Analytics Views and Visuals

SQLQuery2.sql - ac...UAD\nvankay2 (60)* X SQLQuery1.sql - ac...UAD\ nvankay2 (59)*

```
/*NTV M-5*/
/*1.(4) A screenshot from SQL Server Management Studio of the view created by each team member and the data in the view. Show the Create View statement and contents of the view (.pdf file).*/
CREATE VIEW propertiesbylocation
AS
    select propertyid,propertycity
    from PROPERTY;
select *
from propertiesbylocation
order by PropertyCity;
```

Results Messages

	propertyid	propertycity
1	3062548195	Ajo
2	3914768329	Apache Junction
3	6318873767	Apache Junction
4	4747992070	Bouse
5	1475863308	Catalina
6	5729827879	Flagstaff
7	3708287043	Fort Huachuca
8	6298846818	Gilbert
9	4974530133	Gilbert
10	8343867623	Glendale

Query executed successfully. | acadsq17.asurite.ad.asu.edu | ASUAD\ nvankay2 (60) | SP20_CIS365_19668_TEAMS | 00:00:00 | 38 rows

SQLQuery1.sql - a...UAD\mbrewer7 (66)* X

```
/* 1. (4) A screenshot from SQL Server Management Studio of the view created by each team member and the data in the view. Show the Create View statement and contents of the view (.pdf file).
Create a view that produces a fact table from your team's project database

Mat Brewer
STORY: Using Treemap to categorize which Supplier sold the most of each category.*/
CREATE VIEW MatsVisualization AS
SELECT p.PartID, PartCategory, QuantityInStock, ReorderPoint, ReorderQuantity, PartPrice,
       s.SupplierID, SupplierName, SupplierState, DeliveryRating
  FROM PARTS p
 JOIN PARTS_PURCHASED pp
    ON p.PartID = pp.PartID
   JOIN SUPPLIERS s
    ON s.SupplierID = pp.SupplierID;
SELECT *
  FROM MatsVisualization;
```

Results Messages

	PartID	PartCategory	QuantityInStock	ReorderPoint	ReorderQuantity	PartPrice	SupplierID	SupplierName	SupplierState	DeliveryRating
1	0375280087	Compressors	442	78	841	862.17	8061529499	Robert Hubbard	MT	on-time
2	0375280087	Compressors	442	78	841	862.17	8388174425	Amber Young	AR	on-time
3	0547436820	HVAC	104	38	670	952.27	2801844619	John Rollins	GA	on-time
4	0547436820	HVAC	104	38	670	952.27	8192075871	Carrie Ramirez	WV	on-time
5	0818355604	HVAC	278	22	930	238.30	4085239879	Peter Smith	TX	on-time

IC10Visualization.s...SUAD\vsourie (72)*

```

1 /* 1. (4) A screenshot from SQL Server Management Studio of the view created by each team member and the data
2 in the view. Show the Create View statement and contents of the view (.pdf file). Creates a view that produces
3 a fact table from your team's project database.
4
5 Varun Shourie, CIS365, Section 19668 TREE MAP
6 STORY: Using Treemap to list the total hours spent on work orders by residents and properties */
7 CREATE VIEW VarunPropResWO AS
8     SELECT R.ResidentID, R.ResApt, R.ResName, R.ResPhone, R.ResMobilePhone, R.ResEmail, -- RESIDENTS table attributes.
9         WO.WorkOrderID, WO.TaskID, WO.CreatorID, WO.DateCreated, -- WORK ORDER table attributes
10        WO.DesiredWorkDate, WO.StartDate, WO.EndDate, WO.LastMaintDate, WO.Emergency, WO.TotalHours,
11        WO.WorkOrderNotes, WO.WorkOrderStatus,
12        P.PropertyID, P.PropertyName, P.PropertyAddress, P.PropertyCity, P.PropertyState, -- PROPERTY table attributes.
13        P.PropertyZIP, P.PropertyManager, P.PropertyPhone, P.PropertyAccountNo, P.PropertyTech
14    FROM RESIDENTS R JOIN WORK_ORDERS WO
15        ON R.ResidentID = WO.ResidentID
16    JOIN PROPERTY P
17        ON WO.PropertyID = P.PropertyID;
18    SELECT *
19        FROM VarunPropResWO;

```

100 %

	ResidentID	ResApt	ResName	ResPhone	ResMobilePhone	ResEmail	WorkOrderID	TaskID	CreatorID	Date
1	3673914534	410	Sophia Jackson	821-934-2000	103-325-0032	sophia.jackson3@gmail.com	0245421460	7188818037	4723257262	2015
2	5983055224	684	Elizabeth Moore	704-248-0861	808-016-8528	elizabethmoore@gmail.com	1329486777	9599810135	7461707115	2015
3	8270971640	304	Reginald Casey	NULL	846-742-3685	reginaldcasey@gmail.com	1642972264	6694565143	8606515777	2018
4	4557820611	126	Lawrence Curtis	640-498-5677	641-498-5677	lawrence_c_8@gmail.com	1665308421	7684390670	1333527072	2008
5	2263909583	316	Ruth Greene	315-056-4997	NULL	ruthgreene@gmail.com	1858869730	1239259302	3663027200	2020
6	6000000000	100	Laura	502-272-0050	502-272-0050	laura.m.0000000000@gmail.com	21000010050	14000001005	17140501004	2007

Query executed successfully.

SQLQuery1.sql - ac...SUAD\itsanche (71)*

```

/* 1. (4) A screenshot from SQL Server Management Studio of the view created by each team member and the data in the view.
Show the Create View statement and contents of the view (.pdf file). Create a view that produces a fact table from your team's project database

Isis Sanchez - Creating view TechnicianAccidents and join data in Employees, Accidents, and Technicians
Story: Employee Accidents Among Techician Specialties by Severity */

CREATE VIEW TechnicianAccidents AS
    SELECT E.EmpFName, E.EmpLName,
        A.AccID, A.OccurrenceDate, A.AccidentAddress, A.AccidentType, A.WorkSuspDate, A.ResumeDate, A.Status,
        T.TradeSpecialty
    FROM EMPLOYEES E JOIN ACCIDENTS A
        ON E.EmpID = A.EmpID
    JOIN TECHNICIANS T
        ON A.EmpID = T.EmpID;

    SELECT *
        FROM TechnicianAccidents;

```

100 %

	EmpFName	EmpLName	EmpID	AccID	OccurrenceDate	AccidentAddress	AccidentType	WorkSuspDate	ResumeDate	Status	TradeSpecialty
1	Xavier	Michaels	7944061124	1362776825	2011-11-28	1413 S Lynx Dr, Tucson, AZ, 85713	Fell down stairs	2011-11-29	2011-12-02	Minor	HVAC
2	Chad	Wilkerson	4118122171	1647133129	2012-06-18	288 West Mirrorysway, Tempe, AZ, 85280	Fell down stairs	2012-06-19	2012-11-24	Major	Landscaping
3	Raj	Patel	9475027393	2245027268	2017-11-06	777 W Chandler Blvd, Chandler, AZ, 85225	Swallowed ink	2017-11-07	2017-11-10	Minor	Welding
4	Damian	Lillard	0333456789	2292385158	2018-02-04	50809 439th Ave, Wickenburg, AZ, 85390	Attacked by dog	2018-02-05	2018-02-24	Major	Carpentry
5	Dwayne	Wade	0533456789	2305684419	2012-11-07	507 E Queen Creek Dr Apache Junction, AZ, 85218	Car crash	2012-11-08	2012-03-22	Major	Carpentry
6	Alec	Baldwin	0490721952	2596110249	2016-07-07	247 E Gold Dust Way San Tan Valley, AZ, 85143	Fell down ladder	2016-07-08	2016-07-13	Minor	HVAC
7	Greg	McKinney	5368939476	2811356748	2018-03-23	302 N Colorado St, Casa Grande, AZ, 85222	Hand crushed by door	2018-03-24	2018-04-01	Minor	Landscaping
8	John	Mayer	8190360250	2991460024	2018-02-06	728 East Cave Creek Road, Tucson, AZ, 85701	Hit by delivery truck	2018-02-07	2018-11-02	Major	Pools
9	Mark	Ruffalo	3459851464	3510085322	2015-04-06	5646 S Creosote Dr, Apache Junction, AZ, 85218	Fell from ladder	2015-04-07	2015-07-28	Major	Landscaping
10	LeBron	James	0233456789	3538442355	2012-06-20	508 E Silver Reef Rd Casa Grande, AZ, 85222	Car crash	2012-06-22	2012-06-23	Minor	Landscaping
11	Jimmy	Butler	3781003956	3658145182	2014-09-16	563 East Elliot Road, Gilbert, AZ, 85234	Fell from vehicle	2014-09-17	2014-09-19	Minor	Carpentry

Query executed successfully.

SQLQuery1.sql - ac..SUAD\jalewi26 (66)* X

```
/* 1. (4) A screenshot from SQL Server Management Studio of the view created by each team member and the data in the view. Show the Create View statement and contents of the view (.pdf file).  
Create a view that produces a fact table from your team's project database
```

Janae Lewis
STORY: Lists the Work Orders by the number of parts used and property.*/

```
CREATE VIEW WONumberOfPartsUsed AS  
    SELECT W.WorkOrderID,  
           PR.PropertyID, PR.PropertyName,  
           COUNT(PartID) AS "Number of Parts Used"  
      FROM PARTS_USED P JOIN WORK_ORDERS W  
        ON P.WorkOrderID = W.WorkOrderID  
     JOIN PROPERTY PR  
       ON W.PropertyID = PR.PropertyID  
      GROUP BY W.WorkOrderID, PR.PropertyID, PR.PropertyName;  
  
SELECT *  
  FROM WONumberOfPartsUsed;
```

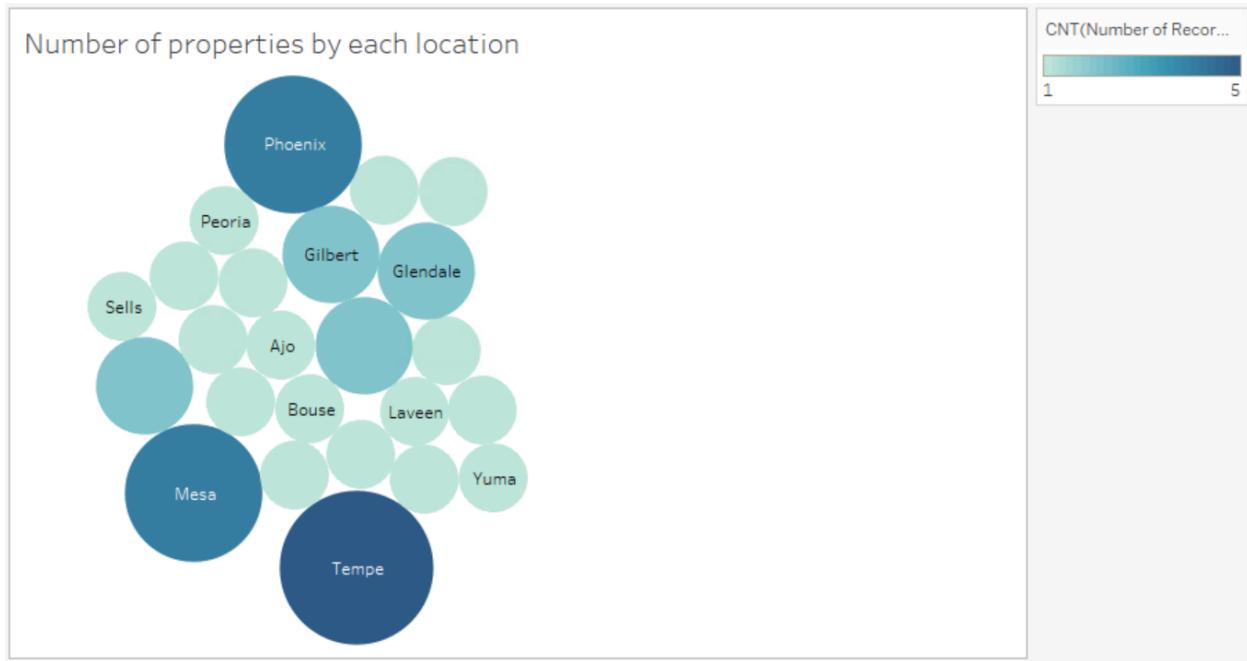
Results Messages

	WorkOrderID	PropertyID	PropertyName	Number of Parts Used
1	0245421460	4974134203	El Paraiso	3
2	1329486777	1778307452	La Paz	2
3	1642972264	2556275703	La Alma	2
4	1665308421	1475863308	Sol Rojo	2
5	1858869730	7175608493	San Salvador	2
6	2168612953	9534546401	Mi Corazon	2
7	2393881879	6616852224	Santiago	2
8	3076766960	9376688927	Playa del Carmen	2
9	3574393747	2908997799	Guadalajara	2
10	4329033923	5395507870	El Pelo	2
11	4727804957	3914768329	Aguas Frescas	2
12	4832636771	2499080739	Las Nubes	2
13	5112788506	2688090757	Bogota	2
14	5120912503	1401958674	Conchas	2
15	5516291197	3199428180	Lima	2
16	5605315456	4399332669	La Casa	2

Query executed successfully. | acadsql17.asurite.ad.asu.edu | ASUAD\jalewi26 (66) | SP20_CIS365_19668_TEAM5 | 00:00:00 | 29 rows

Visualizations

Creator: Nayan Tez Vankalayapati



Creator: Mat Brewer

Parts by Category and Prices with the Supplier Name - Mat Brewer

HVAC Laura Stein	HVAC John Rollins	HVAC Jared Rodriguez	Lubricants Elizabeth Davis	Lubricants Jordan Bryan	Lubricants Barbara Roberson	Pumps Natasha Caldwell	Pumps Thomas Phillips
HVAC Peter Smith							
HVAC Carrie Ramirez	HVAC Rebecca Contreras	HVAC Kari Moore	Lubricants Daniel Larson	Lubricants Gary Jacobs	Lubricants Robert Wells	Pumps Gary Jacobs	Pumps Robert Wells
Blowers Daniel Watson	Blowers Carlos Franco	Blowers Debra Barker	Compressors Amber Young	Compressors Erin Davis	Nuts/Bolts/Screws David Cordova		
Blowers Kathleen Kline	Blowers Carmen Adams	Blowers Tanya Gay	Compressors Robert Hubbard	Compressors Julie Larson	Nuts/Bolts/Screws Richard Shelton		

Creator: Varun Shourie

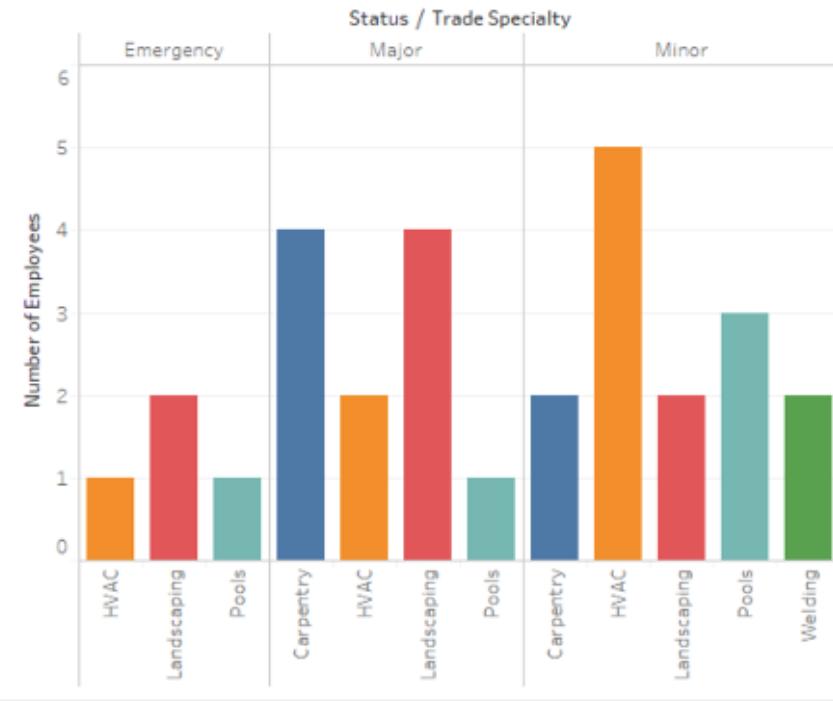
Hours Spent by Property/Resident

<u>El Pelo</u> <u>Jennifer Gardner</u>	<u>Mi Corazon</u> <u>Jamal Brown</u>	<u>El Paraiso</u> <u>Sophia Jackson</u>	<u>La Alma</u> <u>Reginald Casey</u>	<u>Las Flores</u> <u>Marcus Mitchell</u>
	<u>Santiago</u> <u>Zachary Marsh</u>	<u>Las Estrellas</u> <u>Jordan Miller</u>	<u>Las Nubes</u> <u>Lisa</u>	<u>La</u> <u>Luna Azul</u>
		<u>El Lapiz</u> <u>Nicole Johnson</u>	<u>Bogota</u> <u>Kenneth</u>	<u>El</u>
<u>Arroz Con Leche</u> <u>Brandon Pena</u>	<u>Tierra Del Mar</u> <u>Steven Lin</u>	<u>La Casa</u> <u>Kiara Johnson</u>	<u>San</u> <u>Salvador</u>	<u>Los</u> <u>Gatos</u>
	<u>Lima</u> <u>Zachary Evans</u>	<u>La Paz</u> <u>Elizabeth</u>	<u>Vaca</u> <u>Kristina</u>	

Creator: Isis Sanchez

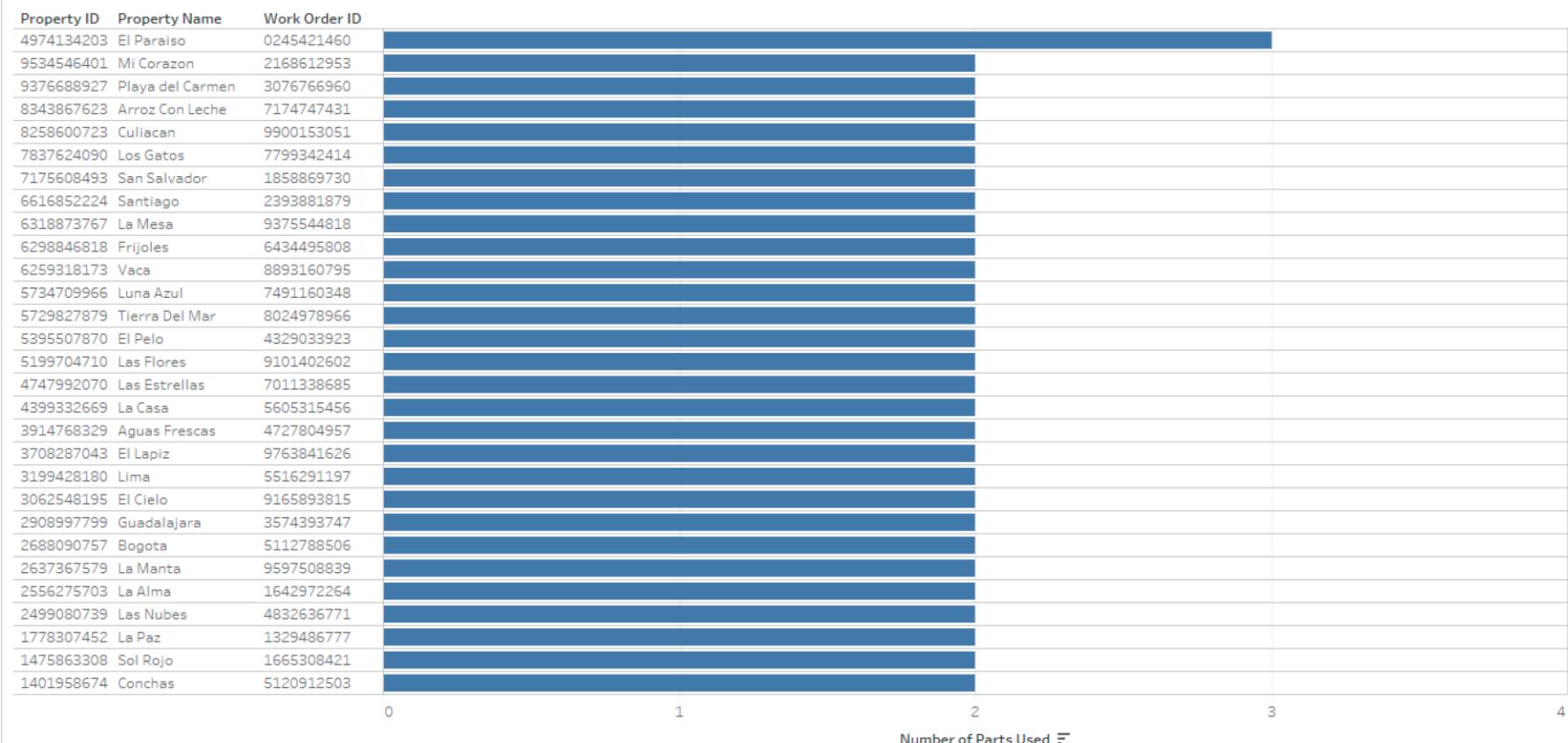
Employee Accidents Among Technician Specialties by Severity

Creator: Isis Sanchez

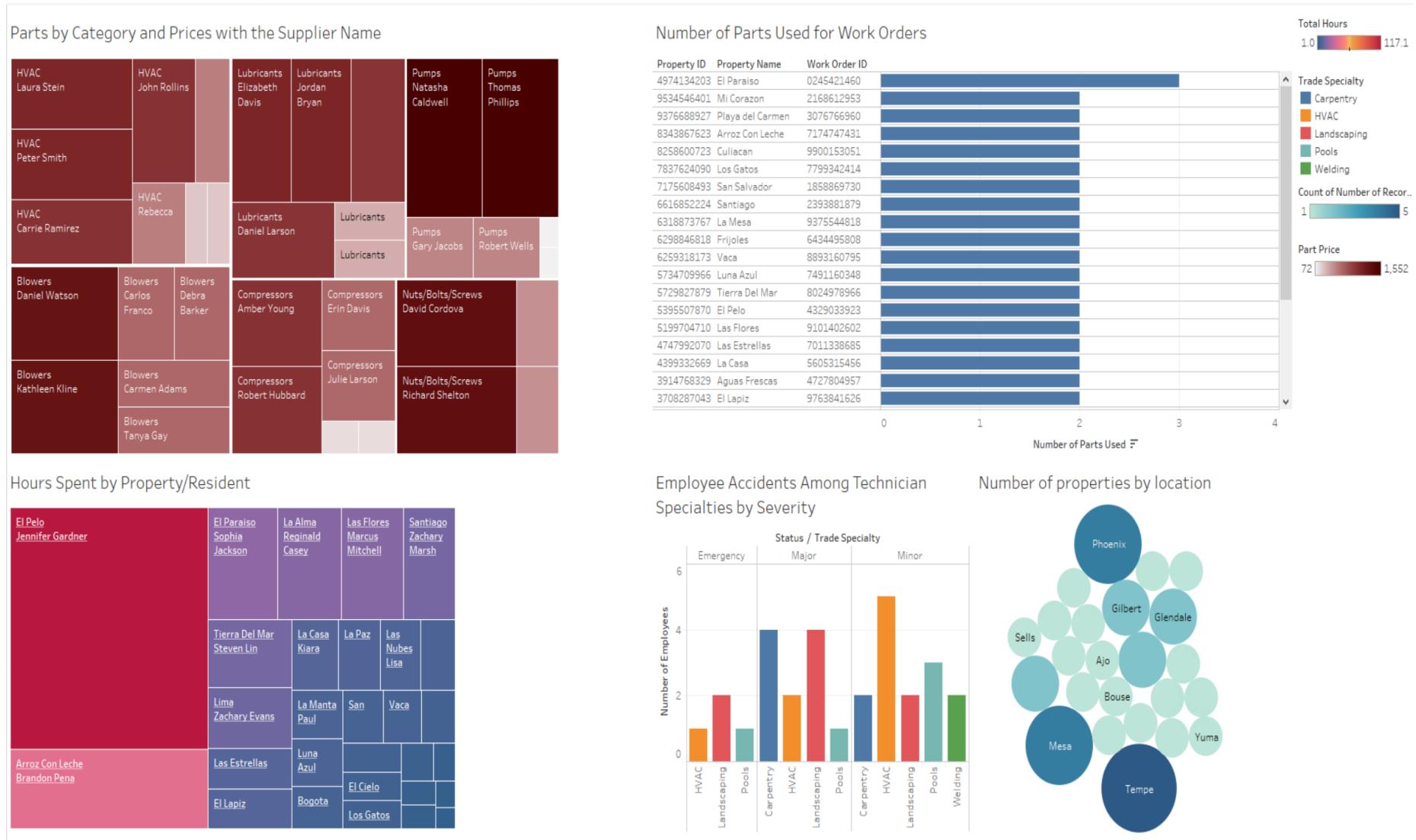


Creator: Janae Lewis

Number of Parts Used for Work Orders



Analytics Dashboard



M5 Revision

/* Linking Employees table to the Users table as requested by the client/professor last minute */

The screenshot shows a SQL query window in SQL Server Management Studio. The code entered is:

```
89  
90  ALTER TABLE USERS  
91      ADD FOREIGN KEY(EmpID)  
92          REFERENCES EMPLOYEES(EmpID);  
93
```

The command is highlighted in blue. Below the code, the message "Command(s) completed successfully." is displayed. At the bottom of the window, a status bar shows "Query executed successfully." and the session details: acadsql17.acad.asean.edu.. ASIAD\vhshoule (66) | SP20_CV5363_VH668_TEAMS | 00:00:00 | 0 rows.

Major Challenges of Database/Project:

During the process of implementing our database, automation of the database, especially surrounding triggers and procedures was difficult. Due to the specifications required by the client, we initially had to slow down and evaluate our options. This step of the process consumed much of our time; we even had to consult with our professor in lieu of our client. Eventually, we always decided on adding a flag attribute which often depicted the status of say, a database user's access or a technician's availability to start on a work order.

Additionally, inserting our tables was difficult because of the lack of usage of source control technologies like GitHub or Microsoft Azure for the code itself. Since code was gathered from group members in one place on a Google Doc, one person had to debug all the minuscule errors while implementing the tables in SQL Server Management Studio

In terms of the database attributes themselves, creating the database design was difficult due to the nature of the client's data. It was a reiterative process in terms of identifying multivalued attributes or potential derived attributes in the database and changing them to associative entities and stored attributes respectively. Working with clients often relies on guesswork, since the sample data provided is often not enough to ascertain the needs of the organization.

Finally, optimizing our database performance proved to be difficult due to the small size of the data set. In order to view more tangible effects of database optimization, we recommend overcoming this problem by inserting more tuples into each relation. Keeping above 100 relations provides an easy outlet to use indexes and nested queries to witness actual effects of query/physical design optimization.

To compound these issues, group members for the latter portion of the project were stuck in different locations in the world--some in India, some in Arizona. One group member absconded from the class also, thereby meaning group members could not reach a timely consensus occasionally.

Lessons Learned:

Our primary piece of advice offered to future database teams is to test changes on a separate table. The simple act of creating a test table with the same constraints and sample data can go a long way in terms of providing a safe valve to test all issues. In case a group member makes a fatal error to database design such as excluding a WHERE clause from the trigger design, the data won't be altered significantly.

Furthermore, learning how to read error messages is a time and skill. Because while resources exist like Stack Overflow to troubleshoot errors, error messages may appear for exceptions outside of its traditional use. For example, while inserting our tables with tuples, we received a notice that our fields had too many characters in them. After inspecting our records, we were dumbfounded initially. However, we deleted one value in the record where the column was problematic. We discovered shortly that there was some weird invisible character which required two backspaces to delete. Therefore, learning how to debug is a must for future projects.

Another important aspect of database design to keep in mind is, do not settle for "good enough." In this project, we had to make important decisions; do we add an ON DELETE CASCADE constraint for an attribute and make deletion of an EMPLOYEE attribute easy, or do we tweak the database design slightly so we can keep data consistency in our database? Keeping our data in the best condition will benefit the client significantly in the future, so all teams should consider this issue when creating, implementing, testing, and modifying their databases.

Furthermore, learning to think in terms of trade-offs is beneficial. Do you or do you not normalize the database, thereby reducing redundancy but also reducing performance? Which attribute do you place indexes on? All of these decisions are something which will cross every database development team's minds which require extensive, well-planned consideration.

APPENDIX

A: SQL

Scripts

APPENDIX A:**All SQL scripts to create your database (Create Table/View, Insert, etc.)**

```

/* 1. Creates USERS Table - Initials: MB, VS*/
Create Table USERS (
    UserID          char(10),
    EmpID          char(10),
    UserPassword   varchar(30) Not Null,
    DateSetUpStamp date        Not Null,
    DateRetired    date
    PRIMARY KEY (UserID),
    CONSTRAINT Date_Check CHECK (DateSetUpStamp <= DateRetired));

/* Inserts data into USERS table */
Insert into USERS values ('0123456789', '1212477095', 'Password', '2019-03-11', '2020-03-11');
Insert into USERS values ('6591784580', '8190360250', 'FavCerealCheerios!', '2019-03-12', '2020-03-12');
Insert into USERS values ('3056923969', '4808236737', 'XB2Ba8BvxLf_9S8Q', '2019-02-01', '2020-03-09');
Insert into USERS values ('4813848184', '2051366864', '&N5ME64rh!42V3!A', '2019-01-16', '2020-03-10');
Insert into USERS values ('1640700292', Null, 'zibd63mBN#an1oC3', '2015-01-07', Null);
Insert into USERS values ('6417891008', '6711164593', '1uyi*PSZRk1OSNuXUYy8k', '2014-01-03', Null);
Insert into USERS values ('0765940686', '4359307467', '2|El8fRDfIQDk-o0^S@K4', '2012-10-29', Null);
Insert into USERS values ('1482310915', '3781003956', 'vk2t78+V2%JDKW&7K|5PL', '2011-10-19', '2020-02-11');
Insert into USERS values ('6405682170', '0485646926', 'aYLL@LxoBR1=2&u3P!ly', '2013-02-01', '2020-03-15');
Insert into USERS values ('4753204707', '4270591322', 'P7_zfKJ&ySCnE4i8HE*a', '2006-03-02', Null);
Insert into USERS values ('7206804872', '1606392601', 'Ta3@JN#2RExR*IzZm8H6', '2013-09-10', Null);
Insert into USERS values ('1845314698', '4118122171', 'CGsmsx+9zTsEv@OzR4Ao', '2009-03-09', Null);
Insert into USERS values ('2777640988', Null, 'pV+84jBDMUGr^Zg*?WKn', '2020-01-01', Null);
Insert into USERS values ('4558228602', Null, 't7t0x=ka#G&pY78%zncU', '2020-03-04', Null);
Insert into USERS values ('8307166856', '8563223391', 'SHXNad+_@-v0UCaBIR0t', '2020-01-09', Null);
Insert into USERS values ('1426569593', '3459851464', '^o6*YXRcO$xQxH+=30y0', '2020-02-12', Null);
Insert into USERS values ('3007276158', '1231737523', 'LEaTGjea+xYJF8g1Obbi', '2018-12-02', Null);
Insert into USERS values ('4513656000', Null, 'VvyF6?3a?%o|OPag3yVG', '2007-09-19', Null);
Insert into USERS values ('6349718771', '2911836367', 'fXP_OjH*1tu', '2013-04-19', Null);
Insert into USERS values ('3715110761', '8149842015', '@6Qin|50SSj', '2014-08-23', Null);
Insert into USERS values ('5238377034', '0490721952', '1xmx@*?V7pb', '2019-03-21', Null);
Insert into USERS values ('9982348171', Null, '6Q96_xTjCom', '2013-03-02', Null);
Insert into USERS values ('4418580110', '5368939476', '*&NTvyTF0c^s', '2014-03-21', Null);
Insert into USERS values ('4995502914', '8830282968', '^4v4Ui5kj^o', '2019-04-03', Null);
Insert into USERS values ('7595427503', '1527728037', 'e_x0EK4!$?kW', '2018-01-05', Null);
Insert into USERS values ('4793796678', '2612804505', '7AkTXQU@b_Jg', '2017-06-09', Null);

```

```

Insert into USERS values ('1441032341', '9475027393', '!6?yPgiTl?', '2014-03-25', Null);
Insert into USERS values ('3269522700', '1712589358', 'Zgq4x*U39e', '2015-02-26', Null);
Insert into USERS values ('8931072793', '7944061124', 'F-4g85bHPW', '2016-05-12', Null);
Insert into USERS values ('0341388010', '5425516460', 'Ir4rbBEj+0_A7WL', '2010-09-12', Null);

```

```
/* Queries data from USERS table */
```

```
SELECT *
```

```
From USERS;
```

```
/* 2. Creates EMPLOYEES table, Initials: MB, VS*/
```

```
Create Table EMPLOYEES (
```

```

EmpID          char(10),
EmpFName      varchar(30) Not null,
EmpLName      varchar(30) Not null,
EmpAddress     varchar(100) Not null,
EmpPhone#      varchar(12) Not null,
EmpJobType    char(1) Not null,
EmgContactName  varchar(60) Not null,
EmgRelationship  varchar(30) Not null,
EmgContactPhone char(12) Not null

```

```
PRIMARY KEY (EmpID),
```

```
CONSTRAINT Phone#_Ck CHECK (EmpPhone# <> EmgContactPhone));
```

```
/* Inserts data into EMPLOYEES table Initials: MB, VS*/
```

```
Insert into EMPLOYEES values ('0123456789', 'John', 'Smith', '1234 East Adams Street, Phoenix, AZ 99921', '333-345-5555', 'T', 'Jane Smith', 'Mother', '222-222-2222');
```

```
Insert into EMPLOYEES values ('1212477095', 'Jack', 'Bosco', '12345 North 69th Avenue, Glendale, AZ 85308', '623-456-9087', 'T', 'Donnie Bosco', 'Father', '810-336-4904');
```

```
Insert into EMPLOYEES values ('8190360250', 'John', 'Mayer', '18886 North 59th Avenue, Glendale, AZ 85308', '623-111-0087', 'T', 'Ellie Mayer', 'Wife', '900-156-1000');
```

```
Insert into EMPLOYEES values ('4808236737', 'Joseph', 'Smith', '10000 Brigham Young Lane, Ogden, UT 34567', '890-345-123', 'T', 'John Smith', 'Father', '954-428-4066');
```

```
Insert into EMPLOYEES values ('2051366864', 'Jay', 'Leno', '15678 Adams Street, Phoenix, AZ 85800', '461-316-7364', 'T', 'Maxwell Leno', 'Father', '713-724-5135');
```

```
Insert into EMPLOYEES values ('6711164593', 'Jim', 'Moran', '8901 West Beaverton Lane, Portland, OR 78902', '802-414-1435', 'T', 'Jesus Moran', 'Father', '890-100-1015');
```

```
Insert into EMPLOYEES values ('4359307467', 'Jusuf', 'Nurkic', '90000 East Beaverton Lane, Portland, OR 78902', '802-789-3456', 'T', 'Hariz Nurkic', 'Father', '424-392-5307');
```

```
Insert into EMPLOYEES values ('3781003956', 'Jimmy', 'Butler', '12345 Chicago Boulevard, Chicago, IL, 34231', '710-241-6433', 'T', 'Rashon Butler', 'Father', '734-718-4418');
```

```
Insert into EMPLOYEES values ('0485646926', 'Django', 'Unchained', '23456 Hocus Pocus Boulevard, Denver, CO 42069', '933-249-7273', 'T', 'Kerry Washington', 'Wife', '924-225-2183');
```

Insert into EMPLOYEES values ('4270591322', 'Quentin', 'Tarantino', '89321 Hollywood Boulevard, Los Angeles, CA 56701', '625-471-7423', 'T', 'Melissa Tarantino', 'Wife', '762-429-1594');

Insert into EMPLOYEES values ('1606392601', 'Brad', 'Pitt', '89300 Hollywood Boulevard, Los Angeles, CA 56701', '869-600-9586', 'T', 'Angelina Jolie', 'Girlfriend', '416-287-4942');

Insert into EMPLOYEES values ('4118122171', 'Chad', 'Wilkerson', '10268 Hollywood Boulevard, Los Angeles, CA 56701', '440-607-5376', 'T', 'Melissa Wilkerson', 'Girlfriend', '683-306-1430');

Insert into EMPLOYEES values ('8563223391', 'Bill', 'Clinton', '89023 Hollywood Boulevard, Los Angeles, CA 56701', '430-600-3535', 'T', 'Hillary Clinton', 'Wife', '524-825-9868');

Insert into EMPLOYEES values ('3459851464', 'Mark', 'Ruffalo', '21890 Hollywood Boulevard, Los Angeles, CA 56701', '313-438-6089', 'T', 'Angela Ruffalo', 'Girlfriend', '666-866-2969');

Insert into EMPLOYEES values ('1231737523', 'Matt', 'Damon', '28901 Hollywood Boulevard, Los Angeles, CA 56701', '952-997-8594', 'T', 'Angie Damon', 'Girlfriend', '893-234-8905');

Insert into EMPLOYEES values ('2911836367', 'Robert', 'Downey', '90231 Hollywood Boulevard, Los Angeles, CA 56701', '191-712-8048', 'T', 'Pepper Potts', 'Wife', '900-678-9023');

Insert into EMPLOYEES values ('8149842015', 'Chris', 'Evans', '69696 Hollywood Boulevard, Los Angeles, CA 56701', '767-778-2502', 'T', 'Krissy Evans', 'Wife', '789-045-7890');

Insert into EMPLOYEES values ('0490721952', 'Alec', 'Baldwin', '90234 Hollywood Boulevard, Los Angeles, CA 56701', '122-680-2312', 'T', 'Alexandra Baldwin', 'Mother', '231-404-8438');

Insert into EMPLOYEES values ('5368939476', 'Greg', 'McKinney', '15890 West Paradise Lane, Glendale, AZ 85308', '382-113-9537', 'T', 'Emma McKinney', 'Wife', '833-794-7360');

Insert into EMPLOYEES values ('8830282968', 'Omar', 'Epps', '90876 West Paradise Lane, Glendale, AZ 85308', '789-729-5330', 'T', 'Olivia Epps', 'Girlfriend', '469-980-1204');

Insert into EMPLOYEES values ('2612804505', 'John', 'Sandusky', '34087 West Paradise Lane, Glendale, AZ 85308', '853-858-7848', 'T', 'Isabella Sandusky', 'Daughter', '125-522-6384');

Insert into EMPLOYEES values ('9475027393', 'Raj', 'Patel', '10101 West Paradise Lane, Glendale, AZ 85308', '229-986-5424', 'T', 'Mona Patel', 'Wife', '890-624-2170');

Insert into EMPLOYEES values ('1712589358', 'Victor', 'Charles', '78902 West Paradise Lane, Glendale, AZ 85308', '453-673-2965', 'T', 'Charlotte Charles', 'Girlfriend', '239-931-3917');

Insert into EMPLOYEES values ('7944061124', 'Xavier', 'Michaels', '45678 West Paradise Lane, Glendale, AZ 85308', '210-727-1706', 'T', 'Mia Michaels', 'Wife', '333-409-2875');

Insert into EMPLOYEES values ('5425516460', 'Oscar', 'Mayer', '97212 West Paradise Lane, Glendale, AZ 85308', '925-449-4190', 'T', 'Amelia Mayer', 'Mother', '847-390-5539');

Insert into EMPLOYEES values ('0133456789', 'Mo', 'Bamba', '89000 North Jackson Street, Nashville, TN 90821', '101-439-5661', 'T', 'Abbie Bamba', 'Girlfriend', '971-449-9562');

Insert into EMPLOYEES values ('0233456789', 'LeBron', 'James', '89021 North Jackson Street, Nashville, TN 90821', '353-243-5081', 'T', 'Savannah James', 'Wife', '808-597-9280');

Insert into EMPLOYEES values ('0333456789', 'Damian', 'Lillard', '89045 North Jackson Street, Nashville, TN 90821', '356-104-5871', 'T', 'Nyjah Lillard', 'Girlfriend', '408-467-5393');

Insert into EMPLOYEES values ('0433456789', 'Bradley', 'Beal', '90782 North Jackson Street, Nashville, TN 90821', '694-885-9647', 'T', 'Chloe McIntosh', 'Girlfriend', '945-680-1191');

Insert into EMPLOYEES values ('0533456789', 'Dwayne', 'Wade', '10234 North Jackson Street, Nashville, TN 90821', '597-334-7635', 'T', 'Gabrielle Union', 'Wife', '141-759-1802');

Insert into EMPLOYEES values ('7834117807', 'Ariana', 'Grande', '10000 South Fountain Avenue, Pasadena, CA, 81003', '223-413-5487', 'O', 'Zayn Malik', 'Boyfriend', '766-512-5214');

Insert into EMPLOYEES values ('4714353184', 'Victoria', 'Justice', '10004 South Fountain Avenue, Pasadena, CA, 81003', '447-512-7373', 'O', 'Nathan Kraft', 'Boyfriend', '725-024-2516');

Insert into EMPLOYEES values ('8161302037', 'Mark', 'Harries', '10001 South Fountain Avenue, Pasadena, CA, 81003', '470-107-5075', 'O', 'Christina Hendricks', 'Girlfriend', '727-681-0854');

Insert into EMPLOYEES values ('1333527072', 'Michelle', 'Obama', '10002 South Fountain Avenue, Pasadena, CA, 81003', '873-425-8235', 'O', 'Barack Obama', 'Husband', '588-165-1870');

Insert into EMPLOYEES values ('6050131166', 'Aamir', 'Khan', '10003 South Fountain Avenue, Pasadena, CA, 81003', '304-203-4114', 'O', 'Aishwarya Rai', 'Wife', '452-187-5872');

Insert into EMPLOYEES values ('8431250187', 'Julia', 'Roberts', '10005 South Fountain Avenue, Pasadena, CA, 81003', '667-408-8800', 'O', 'Tom Hanks', 'Boyfriend', '684-315-4832');

Insert into EMPLOYEES values ('7353470567', 'Chloe', 'Smith', '10006 South Fountain Avenue, Pasadena, CA, 81003', '634-277-3360', 'O', 'Phil Simon', 'Boyfriend', '830-600-0451');

Insert into EMPLOYEES values ('0344221044', 'Aria', 'Johnson', '10007 South Fountain Avenue, Pasadena, CA, 81003', '175-057-5082', 'O', 'Matt Sopha', 'Boyfriend', '865-247-5083');

Insert into EMPLOYEES values ('5051725583', 'Luna', 'Williams', '10008 South Fountain Avenue, Pasadena, CA, 81003', '158-127-2561', 'O', 'Steve Carrell', 'Boyfriend', '180-334-2758');

Insert into EMPLOYEES values ('8576431483', 'Grace', 'Jones', '10009 South Fountain Avenue, Pasadena, CA, 81003', '641-370-7751', 'O', 'Joseph Gordon-Levitt', 'Boyfriend', '207-514-1641');

Insert into EMPLOYEES values ('4058468001', 'Chloe', 'Brown', '10010 South Fountain Avenue, Pasadena, CA, 81003', '202-743-7117', 'O', 'Leonardo DiCaprio', 'Boyfriend', '047-341-6577');

Insert into EMPLOYEES values ('6571006678', 'Adam', 'Davis', '10011 South Fountain Avenue, Pasadena, CA, 81003', '280-076-8824', 'O', 'Sheila Davis', 'Wife', '351-745-8518');

Insert into EMPLOYEES values ('0736351400', 'Zoe', 'Miller', '10012 South Fountain Avenue, Pasadena, CA, 81003', '024-860-2333', 'O', 'Andre Miller', 'Husband', '456-832-5257');

Insert into EMPLOYEES values ('4174185851', 'Leah', 'Wilson', '10013 South Fountain Avenue, Pasadena, CA, 81003', '488-615-5333', 'O', 'Smith Wesson', 'Boyfriend', '527-541-4048');

Insert into EMPLOYEES values ('3455702072', 'Hazel', 'Moore', '10014 South Fountain Avenue, Pasadena, CA, 81003', '110-286-2720', 'O', 'Thomas Magnum', 'Boyfriend', '417-543-8275');

Insert into EMPLOYEES values ('2531566610', 'Violet', 'Taylor', '10015 South Fountain Avenue, Pasadena, CA, 81003', '672-507-6766', 'O', 'Dashiel Hammett', 'Boyfriend', '308-263-5564');

Insert into EMPLOYEES values ('0713445833', 'Aurora', 'Anderson', '10016 South Fountain Avenue, Pasadena, CA, 81003', '321-223-2307', 'O', 'James Bond', 'Boyfriend', '263-854-7765');

Insert into EMPLOYEES values ('7312534106', 'Skylar', 'Thomas', '10017 South Fountain Avenue, Pasadena, CA, 81003', '245-344-7157', 'O', 'Jude Law', 'Boyfriend', '613-605-0708');

Insert into EMPLOYEES values ('5873457506', 'Samantha', 'Harries', '10018 South Fountain Avenue, Pasadena, CA, 81003', '523-468-8145', 'O', 'Kevin Hart', 'Boyfriend', '048-708-2355');

Insert into EMPLOYEES values ('8606515777', 'Maya', 'Jackson', '10019 South Fountain Avenue, Pasadena, CA, 81003', '637-726-8668', 'O', 'Chris Tucker', 'Boyfriend', '811-723-7030');

Insert into EMPLOYEES values ('1184867627', 'Cora', 'White', '10020 South Fountain Avenue, Pasadena, CA, 81003', '337-830-6505', 'O', 'Haroon Elvis', 'Boyfriend', '858-726-8731');

```

Insert into EMPLOYEES values ('6807186162', 'Ivy', 'Harris', '10021 South Fountain Avenue, Pasadena, CA, 81003', '151-130-2434', 'O', 'Matthew Harris', 'Husband', '413-675-3566');
Insert into EMPLOYEES values ('6736040412', 'Sadie', 'Martin', '10022 South Fountain Avenue, Pasadena, CA, 81003', '845-085-4356', 'O', 'Aston Martin', 'Husband', '425-331-0811');
Insert into EMPLOYEES values ('3328752545', 'Delilah', 'Thomson', '10023 South Fountain Avenue, Pasadena, CA, 81003', '174-402-6041', 'O', 'Kobe Thomson', 'Husband', '322-563-6633');
Insert into EMPLOYEES values ('0042412788', 'Arianna', 'Garcia', '10024 South Fountain Avenue, Pasadena, CA, 81003', '230-743-8235', 'O', 'Andre Garcia', 'Husband', '155-520-7776');
Insert into EMPLOYEES values ('7461707115', 'Peyton', 'Martinez', '10025 South Fountain Avenue, Pasadena, CA, 81003', '088-156-3541', 'O', 'Ignacio Martinez', 'Husband', '240-367-6211');
Insert into EMPLOYEES values ('4822341807', 'Mackenzie', 'Robinson', '10026 South Fountain Avenue, Pasadena, CA, 81003', '438-370-4706', 'O', 'Jake Robinson', 'Husband', '140-846-1458');
Insert into EMPLOYEES values ('3663027200', 'Natalia', 'Clark', '10027 South Fountain Avenue, Pasadena, CA, 81003', '658-623-2623', 'O', 'Jamie Clark', 'Husband', '066-506-3120');
Insert into EMPLOYEES values ('1430236724', 'Raelynn', 'Rodriguez', '10028 South Fountain Avenue, Pasadena, CA, 81003', '540-524-1621', 'O', 'Preston Rodriguez', 'Husband', '337-302-5533');
Insert into EMPLOYEES values ('4723257262', 'Kylie', 'Lewis', '10029 South Fountain Avenue, Pasadena, CA, 81003', '878-483-2501', 'O', 'John Lewis', 'Husband', '756-445-3243');

```

/ Queries the EMPLOYEES table */*

Select *

From EMPLOYEES;

/ 3. Creates ACCIDENTS table, Initials: MB, IS*/*

```

Create Table ACCIDENTS(
AccID      char(10),
EmpID      char(10)  Not null,
OccurrenceDate date     Not null,
AccidentAddress varchar(60) Not null,
AccidentType   varchar(30) Not null,
WorkSuspDate  date     Not null,
ResumeDate    date,
WorkOrderNum  char(10) Not null,
Status        varchar(9) Not null,
Primary Key(AccID),
Foreign Key(EmpID) References EMPLOYEES(EmpID));

```

/ Inserts data into ACCIDENTS table */*

```

Insert into ACCIDENTS values ('9472047194', '0123456789', '11-07-2018', '333 North Scottsdale Road, Scottsdale, AZ, 85260', 'Fall from ladder', '11-08-2018', '01-07-2019', '8883412745', 'Major');
Insert into ACCIDENTS values ('5409798359', '1212477095', '10-18-2016', '483 South Desert Way, Gilbert, AZ, 85295', 'Hit by falling toolbox', '10-19-2016', '10-26-2016', '4949152628', 'Minor');

```

Insert into ACCIDENTS values ('2991460024', '8190360250', '02-06-2018', '728 East Cave Creek Road, Tucson, AZ, 85701', 'Hit by delivery truck', '02-07-2018', '11-02-2018', '8551053772', 'Major');

Insert into ACCIDENTS values ('5386717650', '4808236737', '10-12-2019', '374 North Sol Drive, Scottsdale, AZ, 85254', 'Car crash', '10-13-2019', '10-22-2019', '3859250776', 'Minor');

Insert into ACCIDENTS values ('9009326065', '2051366864', '12-14-2017', '316 Bearclaw Road, Payson, AZ, 85541', 'Car crash', '12-15-2017', '03-04-2018', '9995778812', 'Major');

Insert into ACCIDENTS values ('8276777059', '6711164593', '06-09-2020', '929 South Indigo Way, Gilbert, AZ, 85234', 'Burned finger', '06-09-2020', '06-17-2020', '6631017507', 'Minor');

Insert into ACCIDENTS values ('6322192195', '4359307467', '02-24-2015', '212 West Quail Trail, Queen Creek, AZ, 85296', 'Caught in machinery', '02-25-2015', '06-02-2015', '2282064117', 'Major');

Insert into ACCIDENTS values ('3658145182', '3781003956', '09-16-2014', '563 East Elliot Road, Gilbert, AZ, 85234', 'Fell from vehicle', '09-17-2014', '09-19-2014', '9770738170', 'Minor');

Insert into ACCIDENTS values ('6927791567', '0485646926', '01-29-2010', '912 East Hunters Way, Tucson, AZ, 85701', 'Slip on puddle', '02-02-2010', '02-14-2010', '1321165208', 'Minor');

Insert into ACCIDENTS values ('4210463330', '4270591322', '11-17-2011', '305 South Ocean Avenue, Scottsdale, AZ, 85254', 'Car crash', '11-18-2011', '05-25-2012', '8376692785', 'Emergency');

Insert into ACCIDENTS values ('5679555962', '1606392601', '11-22-2011', '192 East Sideview Avenue Chandler, AZ 85794', 'Cut fingertip', '11-23-2011', '11-25-2011', '9977749895', 'Minor');

Insert into ACCIDENTS values ('1647133129', '4118122171', '06-18-2012', '288 West Mirrorsway, Tempe, AZ, 85280', 'Fell down stairs', '06-19-2012', '11-24-2012', '3995754290', 'Major');

Insert into ACCIDENTS values ('5537850105', '8563223391', '01-06-2014', '5635 E La Privada Dr, Cornville, AZ, 86325', 'Car crash', '01-07-2014', '02-03-2014', '4777829942', 'Minor');

Insert into ACCIDENTS values ('3510085322', '3459851464', '04-06-2015', '5646 S Creosote Dr, Apache Junction, AZ, 85218', 'Fell from ladder', '04-07-2015', '07-28-2015', '9570860474', 'Major');

Insert into ACCIDENTS values ('9283964330', '1231737523', '02-04-2019', '5640 E Bell Rd #1078 Scottsdale, AZ, 85254', 'Cut hand', '02-05-2019', '02-13-2019', '3351114777', 'Minor');

Insert into ACCIDENTS values ('5952934898', '2911836367', '12-25-2019', '565 N College Ave Thatcher, AZ, 85552', 'Car crash', '12-27-2019', '07-23-2020', '8248297259', 'Emergency');

Insert into ACCIDENTS values ('6334059225', '8149842015', '06-29-2015', '350 Navajo Trl, Sedona, AZ, 86336', 'Crushed by machinery', '06-29-2015', '02-19-2016', '9292315503', 'Major');

Insert into ACCIDENTS values ('2596110249', '0490721952', '07-07-2016', '247 E Gold Dust Way San Tan Valley, AZ, 85143', 'Fell down ladder', '07-08-2016', '07-13-2016', '8958203970', 'Minor');

Insert into ACCIDENTS values ('2811356748', '5368939476', '03-23-2018', '302 N Colorado St, Casa Grande, AZ, 85222', 'Hand crushed by door', '03-24-2018', '04-01-2018', '3106504382', 'Minor');

Insert into ACCIDENTS values ('6967048856', '8830282968', '07-26-2012', '520 Jordan Rd, Sedona, AZ, 86336', 'Car crash', '07-27-2012', '04-16-2013', '5299748525', 'Major');

Insert into ACCIDENTS values ('6961114481', '2612804505', '12-13-2017', '3390 E Lockett Rd, Flagstaff, AZ, 86004', 'Inhaled toxic fumes', '12-14-2017', '12-20-2017', '8223702874', 'Emergency');

Insert into ACCIDENTS values ('2245027268', '9475027393', '11-06-2017', '777 W Chandler Blvd, Chandler, AZ, 85225', 'Swallowed ink', '11-07-2017', '11-10-2017', '9941356906', 'Minor');

Insert into ACCIDENTS values ('5581334367', '1712589358', '10-22-2013', '3115 N Fairview Ave Tucson, AZ, 85705', 'Car crash', '10-23-2013', '02-17-2014', '8581511497', 'Emergency');

```

Insert into ACCIDENTS values ('1362776825', '7944061124', '11-28-2011', '1413 S Lynx Dr, Tucson, AZ, 85713', 'Fell down stairs', '11-29-2011', '12-02-2011', '1403604079', 'Minor');
Insert into ACCIDENTS values('4343294424', '5425516460', '03-28-2013', '482 S Citrus Ln, Gilbert, AZ, 85297', 'Car crash', '03-28-2013', '08-01-2013', '3764221288', 'Major');
Insert into ACCIDENTS values ('7694579587', '0133456789', '01-31-2012', '507 S Deer Creek Ln Sierra Vista, AZ, 85635', 'Hit by door', '01-31-2012', '02-02-2013', '8857070845', 'Minor');
Insert into ACCIDENTS values ('3538442355', '0233456789', '06-20-2012', '508 E Silver Reef Rd Casa Grande, AZ, 85222', 'Car crash', '06-22-2012', '06-23-2012', '7858463220', 'Minor');
Insert into ACCIDENTS values('2292385158', '0333456789', '02-04-2018', '50809 439th Ave, Wickenburg, AZ, 85390', 'Attacked by dog', '02-05-2018', '02-24-2018', '2823168372', 'Major');
Insert into ACCIDENTS values ('9926071525', '0433456789', '01-08-2018', '50847 Us Hwy #60 89 Wickenburg, AZ, 85390', 'Hit by hammer', '01-08-2018', '03-01-2018', '2029378546', 'Major');
Insert into ACCIDENTS values ('2305684419', '0533456789', '11-07-2012', '507 E Queen Creek Dr Apache Junction, AZ, 85218', 'Car crash', '11-08-2012', '03-22-2012', '6885032822', 'Major');

```

/* Query the entire ACCIDENTS table */

Select *

From ACCIDENTS;

/* 4. Creates TECHNICIANS, Initials: MB*/

```

Create Table TECHNICIANS (
    EmpID      char(10),
    TradeSpecialty  varchar(30) Not null,
    Certificate#   varchar(20) Not null,
    DateOfRenewal   date    Not null,
    DateOfOrigCertificate date    Not null
    PRIMARY KEY(EmpID),
    FOREIGN KEY(EmpID) references EMPLOYEES(EmpID)
);

```

/* Inserts data into technicians table */

```

insert into TECHNICIANS values ('0123456789', 'HVAC', '77821236', '2021-10-10', '2019-10-10');
insert into TECHNICIANS values ('1212477095', 'HVAC', '34378659', '2021-11-19', '2013-11-19');
insert into TECHNICIANS values ('8190360250', 'Pools', '52777873', '2021-09-19', '2015-09-19');
insert into TECHNICIANS values ('4808236737', 'Welding', '73877097', '2020-06-8', '2014-06-08');
insert into TECHNICIANS values ('2051366864', 'HVAC', '5858956', '2020-10-03', '2010-10-03');
insert into TECHNICIANS values ('6711164593', 'Pools', '21615239', '2020-05-09', '2012-05-09');
insert into TECHNICIANS values ('4359307467', 'Landscaping', '58507947', '2020-06-29', '2008-06-29');
insert into TECHNICIANS values ('3781003956', 'Carpentry', '16997861', '2020-08-07', '2010-08-07');
insert into TECHNICIANS values ('0485646926', 'HVAC', '36392361', '2020-12-16', '2012-12-16');
insert into TECHNICIANS values ('4270591322', 'Landscaping', '74620150', '2022-01-13', '2020-1-13');
insert into TECHNICIANS values ('1606392601', 'Pools', '28466386', '2020-11-13', '2004-11-13');

```

```

insert into TECHNICIANS values ('4118122171', 'Landscaping', '45914461', '2021-05-15', '2011-05-15');
insert into TECHNICIANS values ('8563223391', 'HVAC', '71794443', '2012-07-28', '2020-07-28');
insert into TECHNICIANS values ('3459851464', 'Landscaping', '68976384', '2020-11-19', '2012-11-19');
insert into TECHNICIANS values ('1231737523', 'Carpentry', '70459729', '2020-07-12', '2012-07-12');
insert into TECHNICIANS values ('2911836367', 'Pools', '33784841', '2020-12-01', '2002-12-01');
insert into TECHNICIANS values ('8149842015', 'Welding', '30432668', '2021-11-13', '2017-11-13');
insert into TECHNICIANS values ('0490721952', 'HVAC', '76484817', '2020-10-17', '2012-10-17');
insert into TECHNICIANS values ('5368939476', 'Landscaping', '16267565', '2021-02-12', '2007-02-12');
insert into TECHNICIANS values ('8830282968', 'Carpentry', '68174139', '2020-05-21', '2008-05-21');
insert into TECHNICIANS values ('2612804505', 'HVAC', '97344447', '2020-12-16', '2016-12-16');
insert into TECHNICIANS values ('9475027393', 'Welding', '20549261', '2020-06-22', '2016-06-22');
insert into TECHNICIANS values ('1712589358', 'Landscaping', '81186012', '2020-11-28', '2000-11-28');
insert into TECHNICIANS values ('7944061124', 'HVAC', '55115913', '2020-04-04', '2016-04-04');
insert into TECHNICIANS values ('5425516460', 'Carpentry', '8302500', '2020-10-02', '2000-10-02');
insert into TECHNICIANS values ('0133456789', 'Pools', '75288777', '2022-01-23', '2000-05-15');
insert into TECHNICIANS values ('0233456789', 'Landscaping', '97767240', '2020-11-28', '2008-11-28');
insert into TECHNICIANS values ('0333456789', 'Carpentry', '38336214', '2020-05-28', '2015-05-28');
insert into TECHNICIANS values ('0433456789', 'Landscaping', '49066751', '2020-05-16', '2006-05-16');
insert into TECHNICIANS values ('0533456789', 'Carpentry', '79453766', '2020-08-16', '2014-01-06');

```

```

/* Queries the table. */
SELECT *
  From TECHNICIANS;

```

```

/* 5. Creates the OFFICE_STAFF, Initials: MB, VS*/
Create Table OFFICE_STAFF(
EmpID      char(10),
HrsWorkedPerWeek decimal(4,2)  Not null,
HourlyRate   decimal(5,2)  Not null,
PRIMARY KEY(EmpID),
FOREIGN KEY(EmpID) REFERENCES EMPLOYEES(EmpID));

```

```

/* Inserts data into the office staff table */
Insert into OFFICE_STAFF values ('7834117807', 40.00, 13.00);
Insert into OFFICE_STAFF values('4714353184', 27.40, 22.92);
Insert into OFFICE_STAFF values('8161302037', 30.83, 29.99);
Insert into OFFICE_STAFF values('1333527072', 44.90, 35.58);
Insert into OFFICE_STAFF values('6050131166', 31.43, 38.74);
Insert into OFFICE_STAFF values('8431250187', 29.13, 32.82);
Insert into OFFICE_STAFF values('7353470567', 35.69, 18.74);
Insert into OFFICE_STAFF values('0344221044', 35.11, 21.42);
Insert into OFFICE_STAFF values('5051725583', 34.12, 30.04);

```

```

Insert into OFFICE_STAFF values('8576431483', 27.57, 16.72);
Insert into OFFICE_STAFF values('4058468001', 44.72, 22.13);
Insert into OFFICE_STAFF values('6571006678', 46.20, 18.70);
Insert into OFFICE_STAFF values('0736351400', 26.20, 20.98);
Insert into OFFICE_STAFF values('4174185851', 30.27, 22.63);
Insert into OFFICE_STAFF values('3455702072', 40.03, 36.69);
Insert into OFFICE_STAFF values('2531566610', 49.95, 15.73);
Insert into OFFICE_STAFF values('0713445833', 39.32, 30.79);
Insert into OFFICE_STAFF values('7312534106', 42.51, 33.32);
Insert into OFFICE_STAFF values('5873457506', 45.27, 26.34);
Insert into OFFICE_STAFF values('8606515777', 46.81, 20.77);
Insert into OFFICE_STAFF values('1184867627', 36.11, 26.77);
Insert into OFFICE_STAFF values('6807186162', 44.54, 21.35);
Insert into OFFICE_STAFF values('6736040412', 45.09, 30.23);
Insert into OFFICE_STAFF values('3328752545', 25.88, 34.31);
Insert into OFFICE_STAFF values('0042412788', 42.69, 35.54);
Insert into OFFICE_STAFF values('7461707115', 26.32, 37.90);
Insert into OFFICE_STAFF values('4822341807', 38.66, 33.24);
Insert into OFFICE_STAFF values('3663027200', 29.76, 23.08);
Insert into OFFICE_STAFF values('1430236724', 32.13, 35.88);
Insert into OFFICE_STAFF values('4723257262', 26.04, 35.23);

```

/ Queries the entire OFFICE_STAFF table */*

*SELECT **

From OFFICE_STAFF;

/ 6. Creates PARTS Table MB, IS*/*

```

Create Table PARTS (
PartID      char(10),
PartCategory  varchar(30),
PartSubCategory  varchar(30),
PartDescription  varchar(100) Not null,
QuantityInStock  smallint Not null,
SpecialFeatures  varchar(100),
ReorderPoint    smallint Not null,
ReorderQuantity  smallint Not null,
PartPrice      decimal(5, 2) Not null
PRIMARY KEY(PartID));

```

/ Insert data into PARTS table */*

```

Insert into PARTS values('0818355604', 'HVAC', 'Air Quality', 'Air Filter 12x24x2', 278, NULL, 22, 930,
238.30);

```

Insert into PARTS values('0866242420', 'HVAC', 'Air Quality', 'Air Filter 12x24x1', 10, NULL, 80, 390, 858.37);
Insert into PARTS values('3048021228', 'HVAC', 'Air Quality', 'Air Filter 20x20x2', 587, NULL, 79, 9, 56.98);
Insert into PARTS values('0547436820', 'HVAC', 'Air Quality', 'Air Filter 24x24x2', 104, NULL, 38, 670, 952.27);
Insert into PARTS values('2145801385', 'HVAC', 'Air Quality', 'Air Filter 26x30x5', 727, NULL, 46, 888, 152.42);
Insert into PARTS values('5270358105', 'HVAC', 'Air Quality', 'Air Filter 30x40x10', 490, NULL, 59, 931, 402.52);
Insert into PARTS values('4085468765', 'HVAC', 'Air Quality', 'Air Filter 30x20x5', 69, NULL, 58, 852, 233.61);
Insert into PARTS values('8330606618', 'Compressors', '1hp Air Compressor', '2.3Cfm @ 125 Psi, 3 Gal Hotdog, 115V', 406, 'Turbine imported from Germany', 28, 776, 153.02);
Insert into PARTS values('0375280087', 'Compressors', '2hp Air Compressor', '2.4Cfm @ 130 Psi, 4 Gal Hotdog, 120V', 442, 'Turbine imported from Canada', 78, 841, 862.17);
Insert into PARTS values('3534547408', 'Compressors', '3hp Air Compressor', '2.5Cfm @ 135 Psi, 5 Gal Hotdog, 125V', 733, 'Turbine imported from Germany', 50, 916, 149.94);
Insert into PARTS values('6612108073', 'Compressors', '4hp Air Compressor', '2.6Cfm @ 140 Psi, 6 Gal Hotdog, 130V', 39, 'Turbine imported from Japan', 63, 447, 254.99);
Insert into PARTS values('5433565021', 'Compressors', '5hp Air Compressor', '2.7Cfm @ 145 Psi, 7 Gal Hotdog, 135V', 824, 'Turbine imported from Canada', 91, 587, 409.26);
Insert into PARTS values('4756584867', 'Lubricants', 'Compressor Oil, 1 Gal', 'Refrigerator, Compressor Oil', 589, 'Composed of hydrocarbons and alphaolefins', 6, 151, 948.58);
Insert into PARTS values('2373842102', 'Lubricants', 'Compressor Oil, 2 Gal', 'Refrigerator, Compressor Oil', 444, 'Extra viscosity', 49, 81, 154.49);
Insert into PARTS values('7274672886', 'Lubricants', 'Compressor Oil, 3 Gal', 'Refrigerator, Compressor Oil', 628, 'Composed of hydrocarbons and alphaolefins', 76, 371, 479.31);
Insert into PARTS values('8758480004', 'Lubricants', 'Compressor Oil, 4 Gal', 'Refrigerator, Compressor Oil', 20, 'Contains zinc, phosphor, and sulphur', 86, 183, 515.20);
Insert into PARTS values('1356860564', 'Lubricants', 'Compressor Oil, 5 Gal', 'Refrigerator, Compressor Oil', 753, 'Has an alkaline pH', 86, 530, 341.52);
Insert into PARTS values('1665485535', 'Pumps', '1" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', 329, NULL, 32, 493, 520.70);
Insert into PARTS values('2017272650', 'Pumps', '1.5" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', 863, NULL, 74, 418, 66.27);
Insert into PARTS values('3564537651', 'Pumps', '2" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', 201, NULL, 49, 842, 5.67);
Insert into PARTS values('6453146137', 'Pumps', '2.5" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', 407, NULL, 93, 713, 813.61);
Insert into PARTS values('2025818582', 'Pumps', '3" Diaphragm Pump', 'Stainless Steel w/ Santoprene Diaphragm', 723, NULL, 15, 166, 738.87);

```

Insert into PARTS values('4843773686', 'Blowers', 'Belt-Drive Industrial Blowers', '16" W/2 Hp, 1Ph, 115/230v Motor', 702, 'Gasoline-powered', 74, 561, 839.04);
Insert into PARTS values('2477146513', 'Blowers', 'Belt-Drive Industrial Blowers', '17" W/2 Hp, 1Ph, 115/230v Motor', 807, 'Electric battery-powered', 76, 371, 447.54);
Insert into PARTS values('3128304548', 'Blowers', 'Belt-Drive Industrial Blowers', '18" W/2 Hp, 1Ph, 130/245v Motor', 361, 'LPG natural gas powered', 37, 565, 227.39);
Insert into PARTS values('7872520312', 'Blowers', 'Belt-Drive Industrial Blowers', '19" W/2 Hp, 1Ph, 135/250v Motor', 821, 'CNG natural gas powered', 48, 813, 444.58);
Insert into PARTS values('5426810347', 'Blowers', 'Belt-Drive Industrial Blowers', '20" W/2 Hp, 1Ph, 140/255v Motor', 800, 'Hydrogen fuel-cell powered', 29, 592, 671.63);
Insert into PARTS values('4483862646', 'Nuts/Bolts/Screws', 'Bolt', '1/2-13 X 3" Grade 9 Hex Head Bolt', 312, 'Grade 9 Hex Head Bolt', 82, 85, 465.93);
Insert into PARTS values('8850646215', 'Nuts/Bolts/Screws', 'Washer', '1/2-12 Flat Washer', 770, 'FLAT Washer', 44, 147, 585.55);
Insert into PARTS values('1204535835', 'Nuts/Bolts/Screws', 'Screw', '1/4-1-19 Phillips Panhead Sheet Metal Screw', 611, 'Code 51970 functionality', 74, 872, 756.79);

```

/* Queries the PARTS table */

SELECT *

From PARTS;

/* 7. Creates Resident Table, Initials: JL*/

```

Create Table RESIDENTS (
ResidentID  char(10),
ResApt      char(3)  Not null,
ResName     char(60) Not null,
ResPhone    char(12),
ResMobilePhone char(12),
ResEmail    varchar(60),
PRIMARY KEY (ResidentID));

```

/* Inserts data into the Resident table */

```

Insert into RESIDENTS values('3673914534', '410', 'Sophia Jackson', '821-934-2000', '103-325-0032', 'sophiajackson3@gmail.com');
Insert into RESIDENTS values('8715309895', '934', 'Amanda Lopez', '870-540-7164', '870-540-7236', 'amanda.lopez@gmail.com');
Insert into RESIDENTS values('6393256990', '169', 'Jamal Brown', '527-372-6353', '529-372-0093', 'jamal.brown@yahoo.com');
Insert into RESIDENTS values('3780896439', '397', 'Kiara Johnson', '327-544-3537', '326-544-3537', 'kjohnson4@gmail.com');
Insert into RESIDENTS values('4557820611', '126', 'Lawrence Curtis', '640-498-5677', '641-498-5677', 'lawrence_c_8@gmail.com');

```

```
Insert into RESIDENTS values('6080760445', '611', 'Jennifer King', '974-399-1119', '973-399-1119',
'jenniferking@gmail.com');

Insert into RESIDENTS values('7883078832', '305', 'Thomas Miller', '824-690-6153', '825-690-6153',
'1tom.miller@gmail.com');

Insert into RESIDENTS values('7680637430', '122', 'Lisa Andrews', '567-553-9190', '567-553-9190',
'lisa.andrews@hotmail.com');

Insert into RESIDENTS values('1594262954', '523', 'Marcus Mitchell', '275-680-2456', '276-680-2456',
'marc_mitchell@gmail.com');

Insert into RESIDENTS values('7642795175', '438', 'Jacqueline Smith', '893-445-1089', '894-445-1089',
'jackies633l@gmail.com');

Insert into RESIDENTS values('4923659143', '561', 'Steven Lin', '737-718-9661', NULL,
'steve.lin@gmail.com');

Insert into RESIDENTS values('7672339205', '402', 'Sandra Logan', NULL, '885-962-5352',
'sandra_logan@gmail.com');

Insert into RESIDENTS values('9476269763', '328', 'Paul Anderson', '953 218-9581', '603-947-2248',
'p_anderson@gmail.com');

insert into RESIDENTS values('5252053437', '628', 'Brandon Pena', '181-093-9879', NULL,
'brandonpena@gmail.com');

insert into RESIDENTS values('1766091032', '206', 'Nathan Jones', '581-485-4901', '341-483-9189',
'nathanjones@gmail.com');

insert into RESIDENTS values('2640852291', '766', 'Jennifer Smith', '174-340-4593', '750-540-4234',
'jennifersmith@gmail.com');

insert into RESIDENTS values('3498751707', '286', 'Kristina Rangel', '956-337-2629', NULL,
'kristinarangel@gmail.com');

insert into RESIDENTS values('3308976270', '444', 'Glenn Clarke', NULL, '258-575-0306',
'glenncarke@gmail.com');

insert into RESIDENTS values('7253005310', '137', 'Jordan Miller', '522-988-7578', '893-376-6982',
'jordanmiller@gmail.com');

insert into RESIDENTS values('8426508901', '925', 'Nicole Johnson', '205-323-9163', '568-381-0864',
'nicolejohnson@gmail.com');

insert into RESIDENTS values('8270971640', '304', 'Reginald Casey', NULL, '846-742-3685',
'reginaldcasey@gmail.com');

insert into RESIDENTS values('3786184210', '198', 'Jennifer Gardner', '583-911-0103', '847-004-1551',
'jennifergardner@gmail.com');

insert into RESIDENTS values('9599235796', '729', 'Christopher Graves', NULL, '704-459-2825',
'christophergraves@gmail.com');

insert into RESIDENTS values('5210153759', '956', 'Michelle Jordan', '856-668-3005', '130-890-2139',
'michellejordan@gmail.com');

insert into RESIDENTS values('8484639753', '596', 'Dr. William Zimmerman DDS', '042-624-3527', '001-329-6398',
'dr.williamzimmermandds@gmail.com');

insert into RESIDENTS values('5160253592', '284', 'Kenneth Thomas', '838-020-3838', NULL,
'kenneththomas@gmail.com');
```

```

insert into RESIDENTS values('5983055224', '684', 'Elizabeth Moore', '704-248-0861', '808-016-8528',
'elizabethmoore@gmail.com');
insert into RESIDENTS values('8227372978', '441', 'Zachary Evans', NULL, '320-891-1284',
'zacharyevans@gmail.com');
insert into RESIDENTS values('2263909583', '316', 'Ruth Greene', '315-056-4997', NULL,
'ruthgreene@gmail.com');
insert into RESIDENTS values('9003459567', '834', 'Zachary Marsh', '976-096-343', '976-809-9023',
'zacharymarsh@gmail.com');

/* Queries the Resident table */
SELECT *
From RESIDENTS;

/* 8. Creates TASKS table Initials: JL, IS*/
Create Table TASKS(
TaskID      char(10),
TaskType     varchar(30)  Not null,
TaskDescription varchar(100)  Not null,
RootCause    varchar(1000),
PRIMARY KEY(TaskID));

/* Inserts data into TASKS table */
Insert into TASKS values('7897148494', 'Repair', 'Back doors Bldg 102', 'Misuse');
Insert into TASKS values('4423222485', 'Repair', 'Door lock jammed', 'Misuse');
Insert into TASKS values('2987970115', 'Yearly Building Inspection', 'Building Number 101', NULL);
Insert into TASKS values('7684390670', 'Yearly Building Inspection', 'Building Number 101', NULL);
Insert into TASKS values('2463146448', 'C04', 'Cafeteria Water Filters', NULL);
Insert into TASKS values('2813858535', 'C07', 'Cafeteria Ice Machine', NULL);
Insert into TASKS values('7309637974', 'C07', 'Cafeteria Oven', NULL);
Insert into TASKS values('4424071352', 'Yearly Building Inspection', 'Building Number 103', NULL);
Insert into TASKS values('2764771170', 'Repair', 'Room 3', 'Fire damage');
Insert into TASKS values('5475139225', 'C08', 'Emergency Lighting Sys., B/100', NULL);
Insert into TASKS values('4809248271', 'Repair', 'Lawn Tractor', 'Wear/normal use');
Insert into TASKS values('4336837906', 'Fire Hose 1 Year Inspection', 'Fire Hose Cabinet, Bldg 102',
NULL);
Insert into TASKS values('4599572123', 'Repair', 'Cafeteria Refrig & Freezer System', 'Wear/Normal Use');
Insert into TASKS values('2665931830', 'Fire Hose 1 Year Inspection', 'Fire Hose Cabinet, Bldg 102',
NULL);
Insert into TASKS values('7562427718', 'Grounds Maintenance - Weekly', 'Property
Grounds/Landscaping', NULL);
Insert into TASKS values('6374058382', 'Weekly Lobby Inspection', 'Lobby in Building 101', NULL);
Insert into TASKS values('1611618056', 'Pool - Weekly', 'Swimming Pool', NULL);

```

```

Insert into TASKS values('4927352458', 'C04', 'Cafeteria Back Door', NULL);
Insert into TASKS values('6783460537', 'C07', 'Cafeteria Refrigerator', NULL);
Insert into TASKS values('6694565143', 'C08', 'Carrier 400 Ton Chiller, Bldg 101', NULL);
Insert into TASKS values('5993797232', 'C06', 'Elec Traffic Gate - Main Out', NULL);
Insert into TASKS values('6791566934', 'C06', 'Elec Traffic Gate - Main Out', NULL);
Insert into TASKS values('8444985818', 'Grounds Maintenance - Weekly', 'Property Grounds/Landscaping', NULL);
Insert into TASKS values('5826530451', 'Weekly Room Inspection', 'Room 101', NULL);
Insert into TASKS values('2371053656', 'Yearly Building Inspection', 'Building 102', NULL);
Insert into TASKS values('9599810135', 'Repair', 'Emergency Generator', 'Misuse' );
Insert into TASKS values('2749370837', 'Repair', 'Front Parking Lot - 50 spaces', NULL);
Insert into TASKS values('1239259302', 'C06', 'Restroom Pm. Route', NULL);
Insert into TASKS values('4117998742', 'Repair', 'Front Gate', 'car crashed into it');
Insert into TASKS values('7188818037', 'C04', 'Cafeteria Refrigerator', NULL);

/* Queries entire TASKS table */
SELECT *
From TASKS;

/* 9. Creates Property Table, Initials: JL, LS*/
Create Table PROPERTY (
    PropertyID      char(10),
    PropertyName    varchar(30) Not null,
    PropertyAddress varchar(50) Not null,
    PropertyCity    varchar(30) Not null,
    PropertyState   char(2)    Not null,
    PropertyZIP     varchar(10) Not null,
    PropertyManager varchar(30),
    PropertyPhone   char(12)   Not null,
    PropertyAccountNo char(16)  Not null,
    PropertyTech    varchar(30)
    PRIMARY KEY (PropertyID));

/* Inserts data into Property table */
Insert into PROPERTY values('4974134203', 'El Paraiso', '19817 East Camina Plata', 'Queen Creek', 'AZ', '85142', 'Jeanette Pierce', '000-000-0000', '0000000000000001', 'Joshua Gray');
Insert into PROPERTY values('7837624090', 'Los Gatos', '8274 State St', 'Sells', 'AZ', '85634', 'Ella Gonzalez', '245-271-0115', '8414019621139101', 'Patricia Martinez');
Insert into PROPERTY values('9534546401', 'Mi Corazon', '62 Hickory Drive', 'Scottsdale', 'AZ', '85256', 'Jordan Stevens', '774-780-9424', '3754547547606820', 'Daniel Young');
Insert into PROPERTY values('4399332669', 'La Casa', '710 Cactus St.', 'Hualapai', 'AZ', '86412', 'Delaney Hewitt', '242-911-0024', '6781746544571617', 'Louis Moore');

```

Insert into PROPERTY values('1475863308', 'Sol Rojo', '1 Snake Hill Ave.', 'Catalina', 'AZ', '85738', 'Simon Parker', '814-570-4662', '0889826087463648', 'Shawn Watson');

Insert into PROPERTY values('6318873767', 'La Mesa', '884 Walnut Street', 'Apache Junction', 'AZ', '85217', 'Caitlin Huang', '631-687-1580', '6945641282023407', 'Martin Nelson');

Insert into PROPERTY values('1401958674', 'Conchas', '7686 Pennington St.', 'Winkelman', 'AZ', '85192', 'Frederick Smith', '383-228-9193', '9813114865796814', 'Jesse Ramirez');

Insert into PROPERTY values('2499080739', 'Las Nubes', '453 Middle St.', 'Peoria', 'AZ', '85383', 'Anika Jones', '267-345-6676', '2630669620468203', 'Benjamin Parker');

Insert into PROPERTY values('5199704710', 'Las Flores', '494 Seaview St.', 'Patagonia', 'AZ', '85624', 'Theodore Lewis', '526-915-0357', '1753114834766744', 'Sara Stewart');

Insert into PROPERTY values('3914768329', 'Aguas Frescas', '570 S. Summit Ave.', 'Apache Junction', 'AZ', '85220', 'Alexandra Peterson', '677-777-6173', '4270292894585138', 'Kevin Sanders');

Insert into PROPERTY values('5729827879', 'Tierra Del Mar', '9531 Branch Ave.', 'Flagstaff', 'AZ', '86002', 'Jacob Jansen', '665-927-4910', '6123218645323242', 'Anna Griffin');

Insert into PROPERTY values('3062548195', 'El Cielo', '44 Iroquois Dr.', 'Ajo', 'AZ', '85321', 'Veronica Perry', '716-284-8340', '0071558953727405', 'Rebecca Perez');

Insert into PROPERTY values('2637367579', 'La Manta', '61 S. York Rd.', 'Lukeville', 'AZ', '85341', 'Spencer Thompson', '974-621-1582', '1880095808561907', 'Diane Janes');

Insert into PROPERTY values('8343867623', 'Arroz Con Leche', '734 Spruce Dr.', 'Glendale', 'AZ', '85312', 'Annalise Dever', '416-615-9497', '1880871538281439', 'Bruce Carter');

Insert into PROPERTY values('6298846818', 'Frijoles', '891 Oakwood St.', 'Gilbert', 'AZ', '85296', 'Madison Olson', '461-305-7402', '3288208592129698', 'Eric Thompson');

Insert into PROPERTY values('5734709966', 'Luna Azul', '908 Ketch Harbour Dr.', 'Phoenix', 'AZ', '85008', 'Brandon Kean', '677-615-2660', '0959292117477022', 'Gregory White');

Insert into PROPERTY values('6259318173', 'Vaca', '83 Centre Dr.', 'Phoenix', 'AZ', '85065', 'Kelsey Neal', '278-812-6414', '879905650556820', 'Jessica Foster');

Insert into PROPERTY values('8913908162', 'Oro Negro', '10 Jefferson St.', 'Glendale', 'AZ', '85308', 'Mark Curry', '260-931-9309', '5187103738224938', 'Marie Edwards');

Insert into PROPERTY values('4747992070', 'Las Estrellas', '365 Railway Ave.', 'Bouse', 'AZ', '85325', 'Laura Johnson', '689-423-5053', '6983492307000941', 'Deborah Brown');

Insert into PROPERTY values('3708287043', 'El Lapis', '7615 Mechanic Dr.', 'Fort Huachuca', 'AZ', '85670', 'Jack Healy', '848-553-5413', '6876149782170968', 'James Li');

Insert into PROPERTY values('2556275703', 'La Alma', '7024 Chestnut Drive', 'Yuma', 'AZ', '85365', 'Hannah Nguyen', '991-499-3619', '0063621306855729', 'Joe Cook');

Insert into PROPERTY values('5395507870', 'El Pelo', '172 Overlook Avenue', 'Scottsdale', 'AZ', '85256', 'Patrick Henderson', '332-686-8299', '2270604132460494', 'Carl Roberts');

Insert into PROPERTY values('9376688927', 'Playa del Carmen', '54 North Carriage Rd.', 'Laveen', 'AZ', '85339', 'Jessica Rodriguez', '649-383-1372', '3117143816620670', 'Irene Lee');

Insert into PROPERTY values('8258600723', 'Culiacan', '82 Fieldstone St.', 'Phoenix', 'AZ', '85003', 'Julian Martinez', '902-965-8412', '3860750703341019', 'Tammy Wilson');

Insert into PROPERTY values('2908997799', 'Guadalajara', '636 Clark Drive', 'Marble Canyon', 'AZ', '86036', 'Kia Cole', '442-861-3184', '4383634275778438', 'Katrina Diaz');

```

Insert into PROPERTY values('2688090757', 'Bogota', '460 Oak St.', 'Safford', 'AZ', '85548', 'Jacob
Richards', '958-906-8824', '0097191912256572', 'Sonja Norman');
Insert into PROPERTY values('1778307452', 'La Paz', '8887 Anderson Avenue', 'Superior', 'AZ', '85273',
'Alexa Valenzuela', '953-625-9896', '6560393224584714', 'Alejandro Ruiz');
Insert into PROPERTY values('3199428180', 'Lima', '4 Grime St.', 'Tempe', 'AZ', '85281', 'Gabriel Santos',
'698-751-0768', '2207453354499556', 'Lindsay Beck');
Insert into PROPERTY values('7175608493', 'San Salvador', '76 Haven Lane', 'Phoenix', 'AZ', '85097',
'Melissa Garcia', '836-226-4922', '7111551568856522', 'Chelsea Kirk');
Insert into PROPERTY values('6616852224', 'Santiago', '57 Rosemary Lane', 'Mesa', 'AZ', '85207', 'Lamar
Houston', '300-397-7153', '1495262026222291', 'Julia Moran');

```

/ Queries all data from Property Table */*

SELECT *

From PROPERTY;

/ 10. Creates the WORK_ORDERS table, Initials: JL, IS*/*

```

Create Table WORK_ORDERS(
WorkOrderID      char(10),
ResidentID       char(10),
PropertyID       char(10)      Not null,
TaskID           char(10)      Not null,
TechID           char(10),
CreatorID        char(10),
DateCreated      date        Not null,
DesiredWorkDate  date,
StartDate         date,
EndDate          date,
LastMaintDate    date,
Emergency        char(1)      Not null,
TotalHours        Decimal(5, 2) Not null,
WorkOrderNotes   varchar(1000),
WorkOrderStatus  varchar(9)
PRIMARY KEY(WorkOrderID)
FOREIGN KEY(ResidentID) REFERENCES Residents(ResidentID),
FOREIGN KEY(PropertyID) REFERENCES Property(PropertyID),
FOREIGN KEY(TaskID) REFERENCES Tasks(TaskID),
FOREIGN KEY(CreatorID) REFERENCES Employees(EmpID),
FOREIGN KEY(TechID) REFERENCES Technicians(EmpID),
CONSTRAINT DWD_Ck CHECK (DesiredWorkDate >= DateCreated),
CONSTRAINT SD_Ck CHECK (StartDate >= DateCreated),
CONSTRAINT ED_Ck CHECK (EndDate >= StartDate),
CONSTRAINT LMD_Ck CHECK (LastMaintDate >= StartDate),

```

```
CONSTRAINT Emg_Ck CHECK (Emergency = '0' OR Emergency = '1' OR Emergency = '-1'));

/* Inserts data into WORK_ORDERS table */
Insert into WORK_ORDERS values('0245421460', '3673914534', '4974134203', '7188818037',
'0533456789', '4723257262', '2019-11-01', '2019-11-02', '2019-12-03', '2019-12-05', '2019-12-06', '0',
19.00, 'Pump replacement', 'completed');
Insert into WORK_ORDERS values('7799342414', '8715309895', '7837624090', '7897148494',
'0123456789', '7834117807', '2000-08-29', '2000-09-29', '2000-09-29', '2000-09-29', '2000-09-29', '1',
4.00, 'Heater caught fire', 'completed');
Insert into WORK_ORDERS values('2168612953', '6393256990', '9534546401', '4423222485',
'1212477095', '4714353184', '2007-07-06', '2007-08-06', '2007-08-06', '2007-08-06', '2007-08-06', '0',
1.25, 'Broken printer', 'completed');
Insert into WORK_ORDERS values('5605315456', '3780896439', '4399332669', '2987970115',
'8190360250', '8161302037', '2001-07-27', '2001-08-27', '2001-08-27', '2001-08-28', '2001-08-28', '0',
8.00, 'Water filter replacement - cafeteria', 'completed');
Insert into WORK_ORDERS values('1665308421', '4557820611', '1475863308', '7684390670',
'4808236737', '1333527072', '2008-07-30', '2008-08-30', '2008-07-30', '2008-07-30', '2008-07-30', '0',
3.00, 'Lawn Tractor', 'completed');
Insert into WORK_ORDERS values('9375544818', '6080760445', '6318873767', '2463146448',
'2051366864', '6050131166', '2002-06-27', '2002-07-27', '2002-07-27', '2002-07-27', '2002-07-27', '0',
2.00, 'Parking Lot', 'delayed');
Insert into WORK_ORDERS values('5120912503', '7883078832', '1401958674', '2813858535',
'6711164593', '8431250187', '2018-01-14', '2018-02-14', '2018-02-14', '2018-02-14', '2018-02-14', '0',
1.00, 'Door jammed', 'completed');
Insert into WORK_ORDERS values('4832636771', '7680637430', '2499080739', '7309637974',
'4359307467', '7353470567', '2017-04-17', '2017-05-17', '2017-05-17', '2017-05-17', '2017-05-17', '1',
7.00, 'Front Gate', 'Completed');
Insert into WORK_ORDERS values('9101402602', '1594262954', '5199704710', '4424071352',
'3781003956', '0344221044', '2016-10-26', '2016-11-26', '2016-11-26', '2016-11-26', '2016-11-26', '0',
17.00, 'Cafeteria back door repair', 'completed');
Insert into WORK_ORDERS values('4727804957', '7642795175', '3914768329', '2764771170',
'0485646926', '5051725583', '2018-04-23', '2018-05-23', '2018-05-23', '2018-05-23', '2018-05-23', '0',
2.00, 'Speakers repair', 'completed');
Insert into WORK_ORDERS values('8024978966', '4923659143', '5729827879', '5475139225',
'4270591322', '8576431483', '2019-06-14', '2019-07-14', '2019-07-16', '2019-07-17', '2019-07-17', '0',
14.00, 'Bldg. 201 repaint', 'completed');
Insert into WORK_ORDERS values('9165893815', '7672339205', '3062548195', '4809248271',
'1606392601', '4058468001', '2013-02-18', '2013-03-18', '2013-03-18', '2013-03-18', '2013-03-18', '0',
4.00, 'Front Maintenance', 'completed');
Insert into WORK_ORDERS values('9597508839', '9476269763', '2637367579', '4336837906',
'4118122171', '6571006678', '2014-03-17', '2014-04-17', '2014-04-17', '2014-04-17', '2014-04-17', '0',
6.00, 'Front Maintenance', 'completed');
```

```
Insert into WORK_ORDERS values('7174747431', '5252053437', '8343867623', '4599572123',  
'8563223391', '0736351400', '2020-11-01', '2020-12-01', '2020-12-03', '2020-12-06', '2020-12-06', '0',  
38.25, 'Bldg. 102 improvements', 'completed');  
Insert into WORK_ORDERS values('6434495808', '1766091032', '6298846818', '2665931830',  
'3459851464', '4174185851', '2017-03-27', '2017-04-27', '2017-04-28', '2017-04-28', '2017-04-28', '0',  
2.00, 'Room 4 Button Repair', 'completed');  
Insert into WORK_ORDERS values('7491160348', '2640852291', '5734709966', '7562427718',  
'1231737523', '3455702072', '2020-01-02', '2020-02-02', '2020-02-05', '2020-02-05', '2020-02-05', '0',  
6.00, 'Front gate maintenance', 'completed');  
Insert into WORK_ORDERS values('8893160795', '3498751707', '6259318173', '6374058382',  
'2911836367', '2531566610', '2019-12-04', '2020-01-04', '2020-01-05', '2020-01-06', '2020-01-06', '0',  
5.00, 'general maintenance', 'completed');  
Insert into WORK_ORDERS values('8163849853', '3308976270', '8913908162', '1611618056',  
'8149842015', '0713445833', '2019-08-13', '2019-09-13', '2019-09-13', '2019-09-13', '2019-09-13', '0',  
1.30, 'general maintenance', 'completed');  
Insert into WORK_ORDERS values('7011338685', '7253005310', '4747992070', '4927352458',  
'0490721952', '7312534106', '2019-06-17', '2019-07-17', '2019-07-19', '2019-07-19', '2019-07-19', '0',  
8.23, 'air conditioner repair', 'completed');  
Insert into WORK_ORDERS values('9763841626', '8426508901', '3708287043', '6783460537',  
'5368939476', '5873457506', '2010-04-05', '2010-05-05', '2010-05-06', '2010-05-06', '2010-05-06', '0',  
8.21, 'roof repaint', 'completed');  
Insert into WORK_ORDERS values('1642972264', '8270971640', '2556275703', '6694565143',  
'8830282968', '8606515777', '2018-11-14', '2018-12-14', '2018-12-15', '2018-12-16', '2018-12-16', '0',  
17.32, 'Patch ceiling', 'completed');  
Insert into WORK_ORDERS values('4329033923', '3786184210', '5395507870', '5993797232',  
'2612804505', '1184867627', '2016-03-08', '2016-04-08', '2016-04-08', '2016-04-17', '2016-04-17', '0',  
117.05, 'Back room renovation', 'completed');  
Insert into WORK_ORDERS values('3076766960', '9599235796', '9376688927', '6791566934',  
'9475027393', '6807186162', '2017-07-13', '2017-08-13', '2017-08-14', '2017-08-14', '2017-08-14', '0',  
4.17, 'Front Door repair', 'completed');  
Insert into WORK_ORDERS values('9900153051', '5210153759', '8258600723', '8444985818',  
'1712589358', '6736040412', '2016-11-19', '2016-12-19', '2016-12-20', '2016-12-20', '2016-12-20', '0',  
6.00, 'Heater repair', 'completed');  
Insert into WORK_ORDERS values('3574393747', '8484639753', '2908997799', '5826530451',  
'7944061124', '3328752545', '2018-02-28', '2018-03-28', '2018-03-28', '2018-03-28', '2018-03-28', '0',  
4.42, 'Lawn maintenance', 'completed');  
Insert into WORK_ORDERS values('5112788506', '5160253592', '2688090757', '2371053656',  
'5425516460', '0042412788', '2014-05-29', '2014-06-29', '2014-07-04', '2014-07-04', '2014-07-04', '0',  
5.25, 'Cafeteria Oven Inspection', 'completed');  
Insert into WORK_ORDERS values('1329486777', '5983055224', '1778307452', '9599810135',  
'0133456789', '7461707115', '2019-03-18', '2019-04-18', '2019-04-18', '2019-04-18', '2019-04-18', '0',  
7.23, 'Bldg. 102 Inspection', 'completed');
```

```

Insert into WORK_ORDERS values('5516291197', '8227372978', '3199428180', '2749370837',
'0233456789', '4822341807', '2015-01-18', '2015-02-18', '2015-02-20', '2015-02-21', '2015-02-21', '0',
12.44, 'Room 8 Chairs', 'completed');

Insert into WORK_ORDERS values('1858869730', '2263909583', '7175608493', '1239259302',
'0333456789', '3663027200', '2020-07-14', '2020-08-14', '2020-08-22', '2020-08-22', '2020-08-22', '0',
5.23, 'Swimming pool', 'completed');

Insert into WORK_ORDERS values('2393881879', '9003459567', '6616852224', '4117998742',
'0433456789', '1430236724', '2020-03-03', '2020-04-03', '2020-04-06', '2020-04-07', '2020-04-07', '0',
14.15, 'Restroom Room 8', 'Delayed');

/* Queries all data */
SELECT *
FROM WORK_ORDERS;

/* 11. Creates PARTS_USED table - JL, VS */
Create Table PARTS_USED (
WorkOrderID      char(10),
PartID          char(10),
PRIMARY KEY(WorkOrderID, PartID),
FOREIGN KEY(WorkOrderID) REFERENCES WORK_ORDERS(WorkOrderID),
FOREIGN KEY(PartID) REFERENCES PARTS(PartID));

/* Inserts data into Parts_Used table */
Insert into PARTS_USED values ('0245421460', '0818355604');
Insert into PARTS_USED values ('0245421460', '0866242420');
Insert into PARTS_USED values ('7799342414', '0818355604');
Insert into PARTS_USED values ('7799342414', '0866242420');
Insert into PARTS_USED values ('2168612953', '3048021228');
Insert into PARTS_USED values ('2168612953', '0547436820');
Insert into PARTS_USED values ('5605315456', '3048021228');
Insert into PARTS_USED values ('5605315456', '0547436820');
Insert into PARTS_USED values ('1665308421', '2145801385');
Insert into PARTS_USED values ('1665308421', '5270358105');
Insert into PARTS_USED values ('9375544818', '2145801385');
Insert into PARTS_USED values ('9375544818', '5270358105');
Insert into PARTS_USED values ('5120912503', '4085468765');
Insert into PARTS_USED values ('5120912503', '8330606618');
Insert into PARTS_USED values ('4832636771', '4085468765');
Insert into PARTS_USED values ('4832636771', '8330606618');
Insert into PARTS_USED values ('9101402602', '0375280087');
Insert into PARTS_USED values ('9101402602', '3534547408');
Insert into PARTS_USED values ('4727804957', '0375280087');

```

```
Insert into PARTS_USED values ('4727804957', '3534547408');
Insert into PARTS_USED values ('8024978966', '6612108073');
Insert into PARTS_USED values ('8024978966', '5433565021');
Insert into PARTS_USED values ('9165893815', '6612108073');
Insert into PARTS_USED values ('9165893815', '5433565021');
Insert into PARTS_USED values ('9597508839', '4756584867');
Insert into PARTS_USED values ('9597508839', '2373842102');
Insert into PARTS_USED values ('7174747431', '4756584867');
Insert into PARTS_USED values ('7174747431', '2373842102');
Insert into PARTS_USED values ('6434495808', '7274672886');
Insert into PARTS_USED values ('6434495808', '8758480004');
Insert into PARTS_USED values ('7491160348', '7274672886');
Insert into PARTS_USED values ('7491160348', '8758480004');
Insert into PARTS_USED values ('8893160795', '1356860564');
Insert into PARTS_USED values ('8893160795', '1665485535');
Insert into PARTS_USED values ('8163849853', '1356860564');
Insert into PARTS_USED values ('8163849853', '2017272650');
Insert into PARTS_USED values ('7011338685', '3564537651');
Insert into PARTS_USED values ('7011338685', '2017272650');
Insert into PARTS_USED values ('9763841626', '3564537651');
Insert into PARTS_USED values ('9763841626', '6453146137');
Insert into PARTS_USED values ('1642972264', '2025818582');
Insert into PARTS_USED values ('1642972264', '6453146137');
Insert into PARTS_USED values ('4329033923', '2025818582');
Insert into PARTS_USED values ('4329033923', '4843773686');
Insert into PARTS_USED values ('3076766960', '2477146513');
Insert into PARTS_USED values ('3076766960', '4843773686');
Insert into PARTS_USED values ('9900153051', '2477146513');
Insert into PARTS_USED values ('9900153051', '3128304548');
Insert into PARTS_USED values ('3574393747', '7872520312');
Insert into PARTS_USED values ('3574393747', '3128304548');
Insert into PARTS_USED values ('5112788506', '7872520312');
Insert into PARTS_USED values ('5112788506', '5426810347');
Insert into PARTS_USED values ('1329486777', '4483862646');
Insert into PARTS_USED values ('1329486777', '5426810347');
Insert into PARTS_USED values ('5516291197', '4483862646');
Insert into PARTS_USED values ('5516291197', '8850646215');
Insert into PARTS_USED values ('1858869730', '1204535835');
Insert into PARTS_USED values ('1858869730', '8850646215');
Insert into PARTS_USED values ('2393881879', '1204535835');
Insert into PARTS_USED values ('2393881879', '3048021228');
```

```

/* Queries all the data */
SELECT *
  FROM PARTS_USED;

/* 12. Creates Assets Table - JL, IS */
Create Table ASSETS (
AssetID      char(10),
PropertyID   char(10),
AssetSN      varchar(20),
AssetDescript  varchar(100),
AssetType     varchar(60),
AssetLocation  varchar(60),
AssetServiceDate date,
PRIMARY KEY (AssetID),
FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID));

/* Inserts data into Assets Table */
Insert into ASSETS values('0675414264', '4974134203', 'F1345', 'Building Number 101', 'Facility', 'Bldg 3', '2007-11-20');
Insert into ASSETS values('4631604003', '7837624090', 'MAR279', 'Building Number 102', 'Facility', 'Bldg 3', '2010-11-02');
Insert into ASSETS values('0031761327', '9534546401', 'CC2987Y367', 'Cafeteria Water Filter', 'Fixed Equip', 'Bldg 1', '2011-05-30');
Insert into ASSETS values('8175360338', '4399332669', 'P50700200', 'Cafeteria Ice Machine', 'Fixed Equip', 'Bldg 2', '2011-06-06');
Insert into ASSETS values('8104801867', '1475863308', NULL, NULL, NULL, 'Bldg 4', '2014-03-19');
Insert into ASSETS values('7261845041', '6318873767', '873525792XL', 'Cafeteria Oven', 'Fixed Equip', 'Bldg 3', '2002-08-15');
Insert into ASSETS values('0652616070', '1401958674', NULL, NULL, NULL, 'Bldg 5', '2007-01-02');
Insert into ASSETS values('8287642333', '2499080739', NULL, NULL, NULL, 'Bldg 6', '2014-10-22');
Insert into ASSETS values('1753414201', '5199704710', 'GENR13749', 'Cafeteria Refrig & Freezer Sys', 'Fixed Equip', 'Bldg 4', '2015-02-17');
Insert into ASSETS values('8325370104', '3914768329', '2649017147', 'Emergency Lighting Sys, Bldg 102', 'Fire Lighting', 'Bldg 5', '2020-06-29');
Insert into ASSETS values('2770851116', '5729827879', '5718559457SS', 'Emergency Lighting Sys, Bldg 101', 'Fire Lighting', 'Bldg 6', '2013-02-05');
Insert into ASSETS values('4764247258', '3062548195', '7586533', 'Entrance Doors Bldgs. 101, 102', 'Facility', NULL, '2013-04-02');
Insert into ASSETS values('0318380513', '2637367579', 'P70600395z', '1995 Ford F150', 'Vehicle', 'Space 21', '2013-07-31');
Insert into ASSETS values('7311711040', '8343867623', NULL, 'Fire Hose Cabinet, Bldg 101', 'Fire Hose', 'Hallway 1-1', '2014-04-04');

```

```

Insert into ASSETS values('5188651258', '6298846818', NULL, 'Fire Hose Cabinet, Bldg 101', 'Fire Hose',
'Hallway 1-2', '2015-11-13');
Insert into ASSETS values('6121002727', '5734709966', NULL, 'Fire Hose Cabinet, Bldg 102', 'Fire Hose',
'Hallway 2-1', '2004-03-16');
Insert into ASSETS values('6413571648', '6259318173', NULL, 'Fire Hose Cabinet, Bldg 102', 'Fire Hose',
'Hallway 2-1', '2005-06-23');
Insert into ASSETS values('7755135477', '8913908162', '63267234', 'Hand Held Fire Extinguisher, Bldg
102', 'Fire Extinguisher', 'Bldg 102', '2009-07-13');
Insert into ASSETS values('4887684630', '4747992070', '334517089bk', 'Hand Held Fire Extinguisher,
Bldg 101', 'Fire Extinguisher', 'Bldg 101', '2010-07-26');
Insert into ASSETS values('3716163365', '3708287043', 'z4227938974', 'Hand Held Fire Extinguisher',
'Fire Extinguisher', 'Bldg 102', '2019-10-16');
Insert into ASSETS values('3561854783', '2556275703', '10793242', 'Hand Held Fire Extinguisher', 'Fire
Extinguisher', 'Bldg 102', '2003-02-12');
Insert into ASSETS values('4064557878', '5395507870', '6613522754', 'Fire Sprinkler System, Bldg 101',
'Fire Sprinkler', NULL, '2008-03-07');
Insert into ASSETS values('6144830306', '9376688927', 'B387900', 'Fire Sprinkler System, Bldg 102', 'Fire
Sprinkler', NULL, '2013-03-15');
Insert into ASSETS values('3677545754', '8258600723', '9852554420MS', 'Emergency Generator',
'Generator', 'Maintenance Shed', '2016-08-22');
Insert into ASSETS values('7204512486', '2908997799', '99678994', 'Property Grounds/Landscaping',
NULL, NULL, '2017-04-10');
Insert into ASSETS values('0862383767', '2688090757', '9234856184XR', 'Ice Machine', 'Fixed Equip',
'Bldg 101', '2011-03-01');
Insert into ASSETS values('5513614634', '1778307452', NULL, 'Lobby in Building 101', NULL, NULL, '2013-
04-01');
Insert into ASSETS values('5724880563', '3199428180', NULL, 'Front Parking Lot - 50 spaces', 'Facility',
'Front Parking Lot', '2018-07-12');
Insert into ASSETS values('0030025687', '7175608493', NULL, 'Back Parking Lot - 75 spaces', 'Facility',
'Back Parking Lot', '2020-08-12');
Insert into ASSETS values('2874624385', '6616852224', '9052791', 'master sink', NULL, 'Bldg 101', '2018-
05-14');

/* Queries all data from Assets Table */
SELECT *
  From ASSETS;


```

```

/* 13. Creates Suppliers Table, Initials: JL, MB*/
Create Table SUPPLIERS (
SupplierID      char(10),
SupplierName    varchar(30) Not null,
SupplierAddress varchar(100) Not null,
```

```

SupplierCity  varchar(30) Not null,
SupplierState  char(2)    Not null,
SupplierZip   varchar(10) Not null,
SupplierPhone  char(12)   Not null,
SupplierContact varchar(60) Not null,
WebAddress     varchar(200),
DeliveryRating varchar(30),
PRIMARY KEY (SupplierID)
);

```

```

insert into SUPPLIERS values('4085239879', 'Peter Smith', '709 Tina Squares', 'Turnerton', 'TX', '81202',
'535-500-9836', 'Alyssa Fisher', 'Petersmith.com', 'on-time');
insert into SUPPLIERS values('9966475858', 'Laura Stein', '7966 Gray Route Apt. 075', 'Williamsburgh',
'WI', '90083', '503-479-4761', 'Chelsea Cuevas', 'Laurastein.com', 'on-time');
insert into SUPPLIERS values('8192075871', 'Carrie Ramirez', '412 Charles Gateway Suite', 'Brandi', 'WV',
'56145', '902-753-1265', 'William Brown', 'CarrieRamirez.com', 'on-time');
insert into SUPPLIERS values('2801844619', 'John Rollins', '6547 Benitez Streets Suite', 'Halefurt', 'GA',
'06267', '116-746-6015', 'Kristen Burke', 'JohnRollins.com', 'on-time');
insert into SUPPLIERS values('7606421267', 'Rebecca Contreras', '8200 Jessica Extensions Apt. 167',
'Jessica', 'SD', '47582', '959-395-7734', 'Chad Evans', 'RebeccaContreras.com', 'on-time');
insert into SUPPLIERS values('4342523903', 'Jared Rodriguez', '8038 Adams Crest Suite 086',
'Wigginsmouth', 'MI', '25604', '773-289-7132', 'Jeffery Jackson', 'JaredRodriguez.com', 'on-time');
insert into SUPPLIERS values('9522956462', 'Michael Pierce', '66358 Kevin Knoll', 'Christophermouth',
'NJ', '45266', '415-677-9108', 'Joel Robbins', 'MichaelPierce.com', 'on-time');
insert into SUPPLIERS values('5444293005', 'Kari Moore', '71554 Gould Square', 'Andreshire', 'IL',
'06612', '847-865-9292', 'Kevin Williams', 'KariMoore.com', 'on-time');
insert into SUPPLIERS values('8061529499', 'Robert Hubbard', '26075 Chang Trail Suite 743', 'Lake
Monica', 'MT', '48272', '855-965-4915', 'Kara Acosta', 'Roberthubbard.com', 'on-time');
insert into SUPPLIERS values('8388174425', 'Amber Young', '038 Franklin Junction', 'Karenburgh', 'AR',
'70100', '309-591-2183', 'Connie Stafford', 'Amberyoung.com', 'on-time');
insert into SUPPLIERS values('9197677498', 'Julie Larson', '22484 Monica Terrace Suite 181',
'Kristenville', 'NC', '12917', '693-967-6891', 'Sarah Reeves', 'Julielarson.com', 'on-time');
insert into SUPPLIERS values('8951179006', 'Erin Davis', '640 Schaefer Square', 'Johnfort', 'KS', '94029',
'580-386-5372', 'Joseph Larson', 'ErinDavis.com', 'on-time');
insert into SUPPLIERS values('9936615889', 'Elizabeth Davis', '4915 Scott Station', 'Lake Lisaview', 'KY',
'47370', '911-269-6919', 'Christopher Lopez', 'Elizabeth Davis.com', 'on-time');
insert into SUPPLIERS values('5637175929', 'Jordan Bryan', '5684 Henry Land', 'Mckinneyside', 'AL',
'74341', '531-846-6707', 'Kristen Berry', 'JordanBryan.com', 'on-time');
insert into SUPPLIERS values('8956191521', 'Barbara Roberson', '96261 Daniel Camp', 'Rodriguezland',
'CO', '49447', '359-324-7022', 'Nicholas Lang', 'Barbararoberson.com', 'on-time');
insert into SUPPLIERS values('8065159166', 'Daniel Larson', '046 Laura Fields Apt. 982', 'Markhaven',
'MA', '74953', '103-307-8454', 'Kristina Martinez', 'daniellarson.com', 'on-time');

```

```

insert into SUPPLIERS values('2494883889', 'Robert Wells', '6367 Phillips Ways', 'Andreahaven', 'WI',
'89389', '185-829-1563', 'Adrian Simmons', 'Robertwells.com', 'on-time');
insert into SUPPLIERS values('4559406715', 'Gary Jacobs', '47474 Timothy Wells', 'Richardfurt', 'IA',
'61517', '519-488-4322', 'Tony Green', 'garyjacobs.com', 'on-time');
insert into SUPPLIERS values('1381926231', 'Shannon Taylor', 'USNS Carter', 'Scottsdale', 'AZ', '30371',
'298-582-8651', 'Maria Lee', 'shannontaylor.com', 'on-time');
insert into SUPPLIERS values('5275376893', 'Andrew Moore', '524 Robert Mews', 'Raymondchester', 'IA',
'90875', '170-611-2665', 'Amanda Russell', 'Andrewmoore.com', 'damaged');
insert into SUPPLIERS values('2258230140', 'Thomas Phillips', 'USS Rios', 'Phoenix', 'AZ', '11620', '965-
522-8302', 'Matthew Salazar', 'thomasphillips.com', 'on-time');
insert into SUPPLIERS values('3777820462', 'Natasha Caldwell', '7028 McIntyre Ways Port',
'Amandachester', 'AR', '27270', '578-461-9379', 'Eric Brown', 'natashacaldwell.com', 'on-time');
insert into SUPPLIERS values('9387836973', 'Kathleen Kline', 'USNV Golden', 'Santa Monica', 'CA',
'50295', '689-504-6224', 'Daniel Brown', 'kathleenkline.com', 'on-time');
insert into SUPPLIERS values('5238708950', 'Daniel Watson', '3691 Kimberly Union Suite 892',
'Masonville', 'LA', '91729', '728-396-5022', 'Brittney James', 'danielwatson.com', 'on-time');
insert into SUPPLIERS values('9827948951', 'Debra Barker', '34262 Martin Path Apt. 672', 'Steelehaven',
'IN', '58370', '783-512-4068', 'Lisa Trevino', 'debrabarker.com', 'on-time');
insert into SUPPLIERS values('9404798101', 'Carlos Franco', '413 Scott Via', 'Murphyview', 'MD', '66515',
'320-729-7104', 'Andrea Cox', 'carlosfranco.com', 'on-time');
insert into SUPPLIERS values('4803857906', 'Tanya Gay', '139 Joseph Village West', 'Melissaland', 'IA',
'17412', '115-584-7437', 'Melody Bender', 'tanyagay.com', 'on-time');
insert into SUPPLIERS values('6315188977', 'Carmen Adams', '620 Brooks Mountain Apt. 742', 'Lake
Andreaberg', 'NE', '98190', '939-713-9499', 'Kenneth Evans', 'carmenadams.com', 'on-time');
insert into SUPPLIERS values('1037967536', 'David Cordova', '92759 Bennett Curve Apt. 069', 'Leefort',
'OR', '64844', '415-563-7838', 'Amanda Williams', 'davidcordova.com', 'on-time');
insert into SUPPLIERS values('1553882107', 'Richard Shelton', '1311 Martinez Drives Suite 336',
'Grahamview', 'TN', '31408', '451-811-8398', 'Earl Key MD', 'RichardShelton.com', 'on-time');

```

/* Queries all data from Suppliers Table */

SELECT *

From SUPPLIERS;

/* 14. Creates ASSET_SUPPLIER Table, Initials: JL, IS*/

```

Create Table ASSET_SUPPLIER (
SupplierID  char(10),
AssetID     char(10),
PRIMARY KEY (SupplierID, AssetID),
FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID),
FOREIGN KEY(AssetID) REFERENCES Assets(AssetID)
);

```

```
/* Insert data into ASSET_SUPPLIER Table */
Insert into ASSET_SUPPLIER values('4085239879', '0675414264');
Insert into ASSET_SUPPLIER values('4085239879', '4631604003');
Insert into ASSET_SUPPLIER values('9966475858', '0675414264');
Insert into ASSET_SUPPLIER values('9966475858', '4631604003');
Insert into ASSET_SUPPLIER values('8192075871', '0031761327');
Insert into ASSET_SUPPLIER values('8192075871', '8175360338');
Insert into ASSET_SUPPLIER values('2801844619', '0031761327');
Insert into ASSET_SUPPLIER values('2801844619', '8175360338');
Insert into ASSET_SUPPLIER values('7606421267', '8104801867');
Insert into ASSET_SUPPLIER values('7606421267', '7261845041');
Insert into ASSET_SUPPLIER values('4342523903', '8104801867');
Insert into ASSET_SUPPLIER values('4342523903', '7261845041');
Insert into ASSET_SUPPLIER values('9522956462', '0652616070');
Insert into ASSET_SUPPLIER values('9522956462', '8287642333');
Insert into ASSET_SUPPLIER values('5444293005', '0652616070');
Insert into ASSET_SUPPLIER values('5444293005', '1753414201');
Insert into ASSET_SUPPLIER values('8061529499', '8325370104');
Insert into ASSET_SUPPLIER values('8061529499', '1753414201');
Insert into ASSET_SUPPLIER values('8388174425', '8325370104');
Insert into ASSET_SUPPLIER values('8388174425', '2770851116');
Insert into ASSET_SUPPLIER values('9197677498', '4764247258');
Insert into ASSET_SUPPLIER values('9197677498', '2770851116');
Insert into ASSET_SUPPLIER values('8951179006', '4764247258');
Insert into ASSET_SUPPLIER values('8951179006', '0318380513');
Insert into ASSET_SUPPLIER values('9936615889', '7311711040');
Insert into ASSET_SUPPLIER values('9936615889', '0318380513');
Insert into ASSET_SUPPLIER values('5637175929', '7311711040');
Insert into ASSET_SUPPLIER values('5637175929', '5188651258');
Insert into ASSET_SUPPLIER values('8956191521', '6121002727');
Insert into ASSET_SUPPLIER values('8956191521', '5188651258');
Insert into ASSET_SUPPLIER values('8065159166', '6121002727');
Insert into ASSET_SUPPLIER values('8065159166', '6413571648');
Insert into ASSET_SUPPLIER values('2494883889', '7755135477');
Insert into ASSET_SUPPLIER values('2494883889', '6413571648');
Insert into ASSET_SUPPLIER values('4559406715', '7755135477');
Insert into ASSET_SUPPLIER values('4559406715', '4887684630');
Insert into ASSET_SUPPLIER values('1381926231', '3716163365');
Insert into ASSET_SUPPLIER values('1381926231', '4887684630');
Insert into ASSET_SUPPLIER values('5275376893', '3716163365');
Insert into ASSET_SUPPLIER values('5275376893', '3561854783');
```

```

Insert into ASSET_SUPPLIER values('2258230140', '4064557878');
Insert into ASSET_SUPPLIER values('2258230140', '3561854783');
Insert into ASSET_SUPPLIER values('3777820462', '4064557878');
Insert into ASSET_SUPPLIER values('3777820462', '6144830306');
Insert into ASSET_SUPPLIER values('9387836973', '3677545754');
Insert into ASSET_SUPPLIER values('9387836973', '6144830306');
Insert into ASSET_SUPPLIER values('5238708950', '3677545754');
Insert into ASSET_SUPPLIER values('5238708950', '7204512486');
Insert into ASSET_SUPPLIER values('9827948951', '0862383767');
Insert into ASSET_SUPPLIER values('9827948951', '7204512486');
Insert into ASSET_SUPPLIER values('9404798101', '0862383767');
Insert into ASSET_SUPPLIER values('9404798101', '5513614634');
Insert into ASSET_SUPPLIER values('4803857906', '5724880563');
Insert into ASSET_SUPPLIER values('4803857906', '5513614634');
Insert into ASSET_SUPPLIER values('6315188977', '5724880563');
Insert into ASSET_SUPPLIER values('6315188977', '0030025687');
Insert into ASSET_SUPPLIER values('1037967536', '2874624385');
Insert into ASSET_SUPPLIER values('1037967536', '0030025687');
Insert into ASSET_SUPPLIER values('1553882107', '2874624385');
Insert into ASSET_SUPPLIER values('1553882107', '0030025687');

```

```

/* Query all data from ASSET_SUPPLIER Table */
SELECT *
  From ASSET_SUPPLIER;

```

```

/* 15. Creates PARTS_PURCHASED Table, Initials: JL, MS*/
Create Table PARTS_PURCHASED (
PartID      char(10),
SupplierID  char(10),
PRIMARY KEY (PartID, SupplierID),
FOREIGN KEY(PartID) REFERENCES Parts(PartID),
FOREIGN KEY(SupplierID) REFERENCES Suppliers(SupplierID));

```

```

/* Inserts data into PARTS_PURCHASED Table */
Insert into PARTS_PURCHASED values('0818355604', '4085239879');
Insert into PARTS_PURCHASED values('0866242420', '4085239879');
Insert into PARTS_PURCHASED values('0866242420', '9966475858');
Insert into PARTS_PURCHASED values('0818355604', '9966475858');
Insert into PARTS_PURCHASED values('3048021228', '8192075871');
Insert into PARTS_PURCHASED values('0547436820', '8192075871');
Insert into PARTS_PURCHASED values('3048021228', '2801844619');
Insert into PARTS_PURCHASED values('0547436820', '2801844619');

```

```
Insert into PARTS_PURCHASED values('2145801385', '7606421267');
Insert into PARTS_PURCHASED values('5270358105', '7606421267');
Insert into PARTS_PURCHASED values('2145801385', '4342523903');
Insert into PARTS_PURCHASED values('5270358105', '4342523903');
Insert into PARTS_PURCHASED values('4085468765', '9522956462');
Insert into PARTS_PURCHASED values('8330606618', '9522956462');
Insert into PARTS_PURCHASED values('4085468765', '5444293005');
Insert into PARTS_PURCHASED values('8330606618', '5444293005');
Insert into PARTS_PURCHASED values('0375280087', '8061529499');
Insert into PARTS_PURCHASED values('3534547408', '8061529499');
Insert into PARTS_PURCHASED values('0375280087', '8388174425');
Insert into PARTS_PURCHASED values('3534547408', '8388174425');
Insert into PARTS_PURCHASED values('6612108073', '9197677498');
Insert into PARTS_PURCHASED values('5433565021', '9197677498');
Insert into PARTS_PURCHASED values('6612108073', '8951179006');
Insert into PARTS_PURCHASED values('5433565021', '8951179006');
Insert into PARTS_PURCHASED values('4756584867', '9936615889');
Insert into PARTS_PURCHASED values('2373842102', '9936615889');
Insert into PARTS_PURCHASED values('4756584867', '5637175929');
Insert into PARTS_PURCHASED values('2373842102', '5637175929');
Insert into PARTS_PURCHASED values('7274672886', '8956191521');
Insert into PARTS_PURCHASED values('8758480004', '8956191521');
Insert into PARTS_PURCHASED values('7274672886', '8065159166');
Insert into PARTS_PURCHASED values('8758480004', '8065159166');
Insert into PARTS_PURCHASED values('1356860564', '2494883889');
Insert into PARTS_PURCHASED values('1665485535', '2494883889');
Insert into PARTS_PURCHASED values('1356860564', '4559406715');
Insert into PARTS_PURCHASED values('1665485535', '4559406715');
Insert into PARTS_PURCHASED values('2017272650', '1381926231');
Insert into PARTS_PURCHASED values('3564537651', '1381926231');
Insert into PARTS_PURCHASED values('2017272650', '5275376893');
Insert into PARTS_PURCHASED values('3564537651', '5275376893');
Insert into PARTS_PURCHASED values('6453146137', '2258230140');
Insert into PARTS_PURCHASED values('2025818582', '2258230140');
Insert into PARTS_PURCHASED values('6453146137', '3777820462');
Insert into PARTS_PURCHASED values('2025818582', '3777820462');
Insert into PARTS_PURCHASED values('4843773686', '9387836973');
Insert into PARTS_PURCHASED values('2477146513', '9387836973');
Insert into PARTS_PURCHASED values('4843773686', '5238708950');
Insert into PARTS_PURCHASED values('2477146513', '5238708950');
Insert into PARTS_PURCHASED values('3128304548', '9827948951');
Insert into PARTS_PURCHASED values('7872520312', '9827948951');
```

```

Insert into PARTS_PURCHASED values('3128304548', '9404798101');
Insert into PARTS_PURCHASED values('7872520312', '9404798101');
Insert into PARTS_PURCHASED values('5426810347', '4803857906');
Insert into PARTS_PURCHASED values('4483862646', '4803857906');
Insert into PARTS_PURCHASED values('5426810347', '6315188977');
Insert into PARTS_PURCHASED values('4483862646', '6315188977');
Insert into PARTS_PURCHASED values('8850646215', '1037967536');
Insert into PARTS_PURCHASED values('1204535835', '1037967536');
Insert into PARTS_PURCHASED values('8850646215', '1553882107');
Insert into PARTS_PURCHASED values('1204535835', '1553882107');

```

/ Queries data from PARTS_PURCHASED Table */*

```

SELECT *
  From PARTS_PURCHASED;

```

Results Needed by Franklin's Management Services

/ 1. A listing of technicians and their total hours logged for maintenance for each property. */*

```

SELECT t.EmpID, e.EmpFName, e.empLName, p.PropertyID, p.PropertyName,
       SUM(wo.TotalHours) AS "Total Hours Per Property"
  FROM TECHNICIANS t JOIN EMPLOYEES e
    ON t.EmpID = e.EmpID
   JOIN WORK_ORDERS wo
     ON wo.TechID = t.EmpID
   JOIN PROPERTY p
     ON wo.PropertyID = p.PropertyID
 GROUP BY t.EmpID, e.EmpFName, e.empLName, p.PropertyID, p.PropertyName;

```

/ 2. A list of work orders by date completed. */*

```

SELECT *
  FROM WORK_ORDERS wo
 ORDER BY EndDate;

```

/ 3. Property with the most emergency maintenance. */*

```

SELECT TOP 1 p.PropertyID, p.PropertyName, p.PropertyCity, p.PropertyState, p.PropertyZIP,
       (SUM(wo.TotalHours)) AS "Total Hours of Emergency Maintenance Performed" -- Cannot use
                                -- MAX(SUM(wo.TotalHours))
  FROM PROPERTY p JOIN WORK_ORDERS wo
    ON p.PropertyID = wo.PropertyID
   WHERE wo.Emergency = '1'

```

```
GROUP BY p.PropertyID, p.PropertyName, p.PropertyCity, p.PropertyState,
p.PropertyZIP
```

ORDER BY 6 DESC; -- By ordering by the sum quantity in descending order and selecting the TOP 1,
-- we can bypass the limitation with not being able to use MAX(SUM(wo.TotalHours))

```
SELECT * FROM PROPERTY;
```

```
SELECT * FROM WORK_ORDERS;
```

/* 4. The technicians who must renew their trade certificates in the next month. */

```
SELECT t.EmpID, e.EmpFName, e.EmpLName,
      t.DateOfRenewal AS "Trade Certificate Renewal Date"
   FROM TECHNICIANS t JOIN EMPLOYEES e
      ON t.EmpID = e.EmpID
     WHERE DATEDIFF(MONTH, GETDATE(), DateOfRenewal) <= 1;
```

/* 5. Which unit required the most maintenance in the past year? */

```
SELECT TOP 1 r.ResidentID, r.ResApt,
      p.PropertyID, p.PropertyName, p.PropertyCity, p.PropertyState, p.PropertyZIP,
      (SUM(wo.TotalHours)) AS "Total Hours of Maintenance Performed" -- Can't use
MAX(SUM(wo.TotalHours))
   FROM PROPERTY p JOIN WORK_ORDERS wo
      ON p.PropertyID = wo.PropertyID
   JOIN RESIDENTS r
      ON wo.ResidentID = r.ResidentID
     WHERE DATEDIFF(YEAR, wo.LastMaintDate, GETDATE()) <= 1
          GROUP BY r.ResidentID, r.ResApt, p.PropertyID, p.PropertyName,
                  p.PropertyCity, p.PropertyState, p.PropertyZIP
```

ORDER BY 8 DESC; -- By ordering the sum quantity in descending order and selecting the TOP 1,
-- we can bypass the limitation with not being able to use MAX(SUM(wo.TotalHours))

/* 6. Which unit required no maintenance in the past year? */

```
SELECT r.ResidentID, r.ResApt, p.PropertyID, p.PropertyName, p.PropertyCity,
      p.PropertyState, p.PropertyZIP
   FROM PROPERTY p JOIN WORK_ORDERS wo
      ON p.PropertyID = wo.PropertyID
   JOIN RESIDENTS r
      ON r.ResidentID = wo.ResidentID
     WHERE p.PropertyID NOT IN -- Excludes properties which have been serviced in
the past
          (SELECT pr.PropertyID -- year in case properties have multiple work
orders.
          FROM PROPERTY pr JOIN WORK_ORDERS wo2
```