

## **IPA 3**

**Name- Akshai Seshadri**  
**UTD ID- 2021547613**

1) **Customer Table**

SELECT \* FROM Customer

182    SELECT \* FROM Customer

	CustID	CustName	CustZipcode
1	101	GILMORE	75252
2	102	Waters	95032
3	103	Harrison	75280
4	104	Moon	98070
5	105	Townsend	75252

## 2) Reports Table

SELECT \* FROM Reports

183  
184    SELECT \* FROM Reports  
185

	ReportID	Test_Date	Validity	Observations	StdRange
1	411	2018-09-14	2020-04-03	234	250
2	420	2019-10-24	2020-04-23	323	400
3	444	2020-02-02	2020-08-23	564	560
4	690	2019-01-31	2019-07-30	646	670
5	701	2018-04-08	2018-10-08	234	250
6	720	2019-09-01	2019-03-31	786	900
7	723	2018-05-14	2018-12-14	2414	2500
8	735	2018-09-14	2019-03-14	474	520
9	892	2018-12-26	2019-05-25	131	150
10	920	2019-04-14	2019-10-13	132	160

## 3) Equipment Table

SELECT \* FROM Equipment

## 4) Supplier Table

SELECT \* FROM Supplier

```

185
186 SELECT * FROM Equipment
187

```

Results		Messages
	PartID ▾	PartName ▾
1	123	Sprocket
2	234	Gear
3	420	Spur
4	421	Helic
5	516	Mesh
6	537	Fishbone
7	846	Calliper
8	920	Fastener
9	987	Harness

```

187
188 SELECT * FROM Supplier
189

```

Results		Messages
	SupplierID ▾	SupplierName ▾
1	175	Schlumberger
2	632	Minda
3	678	Bosch
4	900	Siemens

#### 5) Inspector Table

SELECT \* FROM Inspector

#### 6) Insurance Table

SELECT \* FROM Insurance

189		
190	SELECT * FROM Inspector	
191		
Results Messages		
	InspectorID ▾	InspectorName ▾
1	201	John
2	210	Mike
3	252	Solomon
4	270	Doron
5	281	Jacob
191		
192	SELECT * FROM Insurance	
193		
Results Messages		
	PolicyID ▾	PolicyName ▾
1	72	Titan
2	80	Gold
3	82	Platinum
4	88	Mercury
5	89	Company
6	91	Swaroski
7	92	Tanishq
8	93	Diamond
9	94	Tiffany

7) Plans Table

SELECT \* FROM Plans

8) Cases Table

SELECT \* FROM Cases

```

193
194 SELECT * FROM Plans
195

```

Results		Messages
	PlanID	PlanType
1	1	Mango
2	2	test 7
3	3	Tomatillo
4	4	test 4
5	5	China
6	6	Britan
7	7	test 7
8	8	Squash

```

195
196 SELECT * FROM Cases
197

```

Results		Messages
	CaseID	TestName
1	10	Emission
2	11	Emission
3	20	Acid PM 2
4	21	Acid PM 2
5	22	Litmus test
6	33	Base Emission
7	34	Base Emission
8	42	Solvent
9	43	Solvent
10	50	PM 10
11	51	PM 10
12	70	N20
13	71	N20
14	80	GCarsinogen

## 9) Details Table

SELECT \* FROM Details

203

204 SELECT \* FROM Details

205

Results

Messages

	CustID ▾	ReportID ▾	InspectorID ▾	PolicyID ▾	PlanID ▾	CaseID ▾	PartID ▾	SupplierID ▾
1	101	701	201	80	6	20	123	632
2	101	723	281	80	6	21	123	632
3	101	735	201	82	4	42	234	678
4	101	892	201	82	4	43	234	690
5	102	690	281	93	3	33	516	175
6	102	920	201	93	8	80	516	175
7	102	720	210	94	3	34	537	175
8	103	420	281	88	1	10	420	900
9	105	444	201	92	2	22	987	632
1...	102	720	210	94	3	34	537	175
1...	101	701	201	80	6	20	123	632
1...	101	723	281	80	6	21	123	632
1...	101	735	201	82	4	42	234	678
1...	101	892	201	82	4	43	234	690
1...	102	690	281	93	3	33	516	175
1...	102	920	201	93	8	80	516	175
1...	102	720	210	94	3	34	537	175
1...	103	420	281	88	1	10	420	900
1...	105	444	201	92	2	22	987	632
2...	103	411	270	89	1	11	421	900

## 10) Inner Join-

```
1) /* Inner Joint */
2)
3) Select Details.CustID, Customer.CustID, Customer.CustName, Customer.CustZipcode,
   Reports.Test_Date, Reports.ReportID, Reports.Observations, Reports.StdRange
4) from Customer INNER JOIN Details
5) on Customer.CustID=Details.CustID INNER JOIN Reports
6) on Details.ReportID=Reports.ReportID
7) Where Details.CustID = '101'
```

Results Messages									
	CustID	CustID	CustName	CustZipcode	Test_Date	ReportID	Observations	StdRange	
1	101	101	GILMORE	75252	2018-04-08	701	234	250	
2	101	101	GILMORE	75252	2018-05-14	723	2414	2500	
3	101	101	GILMORE	75252	2018-09-14	735	474	520	
4	101	101	GILMORE	75252	2018-12-26	892	131	150	
5	101	101	GILMORE	75252	2018-04-08	701	234	250	
6	101	101	GILMORE	75252	2018-05-14	723	2414	2500	
7	101	101	GILMORE	75252	2018-09-14	735	474	520	
8	101	101	GILMORE	75252	2018-12-26	892	131	150	

The Function **INNER JOIN** selects records that have matching values in both tables.

Using the inner join, we have fetched the Test **Reports** that the **inspector/ agency generated** and **Customer details** of the customer having customer id as “101”.

```
205
206 /* Inner Joint */
207
208 select Details.CustID, Customer.CustID, Customer.CustName, Customer.CustZipcode, Reports.Test_Date, Reports.ReportID, Reports.Observations, Reports.StdRange
209 from Customer INNER JOIN Details
210 on Customer.CustID=Details.CustID INNER JOIN Reports
211 on Details.ReportID=Reports.ReportID
212 where Details.CustID = '101'
```

Results

Messages

	CustID	CustID	CustName		CustZipcode		Test_Date	ReportID	Observations	StdRange
1	101	101	GILMORE	...	75252	...	2018-04-08	701	234	250
2	101	101	GILMORE	...	75252	...	2018-05-14	723	2414	2500
3	101	101	GILMORE	...	75252	...	2018-09-14	735	474	520
4	101	101	GILMORE	...	75252	...	2018-12-26	892	131	150
5	101	101	GILMORE	...	75252	...	2018-04-08	701	234	250
6	101	101	GILMORE	...	75252	...	2018-05-14	723	2414	2500
7	101	101	GILMORE	...	75252	...	2018-09-14	735	474	520
8	101	101	GILMORE	...	75252	...	2018-12-26	892	131	150

## 11) Outer Join-

```
/* Outer Joint */
```

```
Select Reports.ReportID, Reports.Test_Date, Reports.Observations,  
Inspector.InspectorName, Insurance.PolicyName  
from Inspector FULL OUTER JOIN Reports  
on Inspector.InspectorID=Reports.ReportID FULL OUTER JOIN Insurance  
on Reports.ReportID=Insurance.PolicyName
```

Results		Messages				
	ReportID	Test_Date	Observations	InspectorName	InspectorName	PolicyName
1	NULL	NULL	NULL	John	John	NULL
2	NULL	NULL	NULL	Mike	Mike	NULL
3	NULL	NULL	NULL	Solomon	Solomon	NULL
4	NULL	NULL	NULL	Doron	Doron	NULL
5	NULL	NULL	NULL	Jacob	Jacob	NULL
6	125	2020-08-08	233	NULL	NULL	NULL
7	411	2018-09-14	234	NULL	NULL	NULL
8	420	2019-10-24	323	NULL	NULL	NULL
9	444	2020-02-02	564	NULL	NULL	NULL
1...	666	2020-03-02	232	NULL	NULL	NULL
1...	690	2019-01-31	646	NULL	NULL	NULL
1...	701	2018-04-08	234	NULL	NULL	NULL
1...	720	2019-09-01	786	NULL	NULL	NULL
1...	723	2018-05-14	2414	NULL	NULL	NULL
1...	735	2018-09-14	474	NULL	NULL	NULL
1...	762	2020-05-31	786	NULL	NULL	NULL
1...	891	2020-04-04	576	NULL	NULL	NULL
1...	892	2018-12-26	131	NULL	NULL	NULL
1...	920	2019-04-14	132	NULL	NULL	NULL

The function 'FULL OUTER JOIN' returns all records when there is a match in left or right table records. In this case, Full Join is used to fetch all the records of **Inspector** and **Insurance details** of the **Reports** generated if and when corresponding values are present.

## 12) Left Outer Join

```
13)
14) Select Cases.CaseID,Cases.TestName,Plans.PlanType,Plans.PlanID
15) from Plans LEFT OUTER JOIN Details
16) on Plans.PlanID=Details.PlanID LEFT OUTER JOIN Cases
17) on Details.CaseID=Cases.CaseID
```

**Results**   **Messages**

	CaseID ▾	TestName ▾	PlanType ▾	PlanID ▾
1	10	Emission	Mango	1
2	10	Emission	Mango	1
3	11	Emission	Mango	1
4	22	Litmus test	test 7	2
5	22	Litmus test	test 7	2
6	22	Litmus test	test 7	2
7	22	Litmus test	test 7	2
8	33	Base Emission	Tomatillo	3
9	34	Base Emission	Tomatillo	3
10	34	Base Emission	Tomatillo	3
11	33	Base Emission	Tomatillo	3
12	34	Base Emission	Tomatillo	3
13	42	Solvent	test 4	4

The function 'Left out join' returns all records from the left table, and the matched records from the right table. In cases where there is no match, the result is NULL.

Above, left outer join fetches all the **Plan details** and **case details**.



### 13)Right Outer Join

```
Select Supplier.SupplierID, Supplier.SupplierName, Equipment.PartID,  
Equipment.PartName  
from Supplier LEFT OUTER JOIN Details  
on Supplier.SupplierID =Details.SupplierID LEFT OUTER JOIN Equipment  
on Details.PartID=Equipment.PartID
```

Results		Messages		
Results grid				
	SupplierID ▾	SupplierName ▾	PartID ▾	PartName ▾
1	175	Schlumberger	516	Mesh
2	175	Schlumberger	516	Mesh
3	175	Schlumberger	537	Fishbone
4	175	Schlumberger	537	Fishbone
5	175	Schlumberger	516	Mesh
6	175	Schlumberger	516	Mesh
7	175	Schlumberger	537	Fishbone
8	632	Minda	123	Sprocket
9	632	Minda	123	Sprocket
10	632	Minda	987	Harness
11	632	Minda	123	Sprocket
12	632	Minda	123	Sprocket
13	632	Minda	987	Harness

The function 'Right outer join' returns all records from the right table, and the matched records from the left table. In cases where there is no match, the result is NULL.

Above, function Right outer join fetches all the **Supplier** and **equipment** details.

### 14)Intersect

```
232
233  /*Intersect*/
234
235  Select CustID from Customer
236  INTERSECT
237  Select CustID from Details
238
```

---

Results

Messages

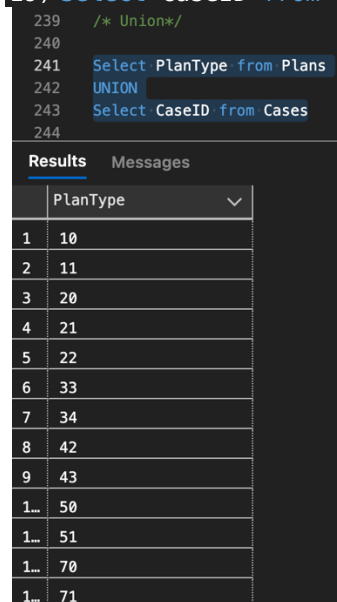
	CustID ▾
1	101
2	102
3	103
4	104
5	105

As the function 'INTERSECT' combines result sets of two or more queries and returns distinct rows that are output by both queries.

In the instance above, we are intersecting the results the CUSTID of Customers who also have Reports.

## 15) Union

```
16)
17) Select PlanType from Plans
18) UNION
19) Select CaseID from Cases
```



The screenshot shows a SQL query editor with a dark background. The query is as follows:

```
239  /* Union*/
240
241  Select PlanType from Plans
242  UNION
243  Select CaseID from Cases
244
```

Below the query, there is a 'Results' tab selected, showing a table with the following data:

	PlanType
1	10
2	11
3	20
4	21
5	22
6	33
7	34
8	42
9	43
1...	50
1...	51
1...	70
1...	71

Union combines results of two SELECT statements into a single result.

Above, the function **Union** creates a collection of all the **Case IDs** from **Cases table** and **Plan table**.

## 16) Except

```
Select ReportID from Reports
EXCEPT
Select InspectorID from Details
```

Results		Messages
	ReportID ▾	
1	125	
2	411	
3	420	
4	444	
5	666	
6	690	
7	701	
8	720	
9	723	
10	735	
11	762	
12	891	
13	892	

EXCEPT subtracts the result set of a query from another and gives distinct results.  
Above, we find all **Reports** who do not have corresponding **Inspectors**.