

ASSIGNMENT :-

Q1) Display ItemCode, ItemName and ItemCategory of items whose UnitPrice is less than INR 500.

Ans.

```
mysql> Select ItemCode, ItemName, ItemCategory from item where UnitPrice < 500;
+-----+-----+-----+
| ItemCode | ItemName | ItemCategory |
+-----+-----+-----+
| IT105    | Pencils  | Stationary   |
| IT110    | Eraser   | Stationary   |
| IT115    | Pens     | Stationary   |
| IT120    | Sharpner | Stationary   |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Q2. Display ItemCode, ItemName and ItemCategory of items whose UnitPrice is in the range INR 10,000 to INR 20,000 (both inclusive).

```
mysql> Select ItemCode, ItemName, ItemCategory,UnitPrice from item where UnitPrice between '10000' and '20000';
+-----+-----+-----+-----+
| ItemCode | ItemName      | ItemCategory | UnitPrice |
+-----+-----+-----+-----+
| IT101    | LED32Inch     | Television   | 15000.00 |
| IT102    | GalaxyS5      | Mobile       | 13000.00 |
| IT103    | FrontLoad     | WashingMachine | 20000.00 |
| IT104    | DualDoor      | Refrigerator | 12000.00 |
| IT106    | LED32Inch     | Television   | 19000.00 |
| IT107    | LGG5Silver    | Mobile       | 11000.00 |
| IT109    | FourDoor      | Refrigerator | 18000.00 |
| IT114    | ThreeDoorNormal | Refrigerator | 19000.00 |
| IT116    | LED32Inch     | Television   | 20000.00 |
| IT117    | Onidai505     | Mobile       | 10000.00 |
| IT118    | SemiAutomatic | WashingMachine | 11000.00 |
| IT119    | ThreeDoorLux  | Refrigerator | 18000.00 |
+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

Q3. Display ItemCode and ManufacturerName of Televisions that are costing more than INR 15,000 .

```
mysql> Select ItemCode, ManufacturerName from item where ItemCategory='Television' and UnitPrice > 15000;
+-----+-----+
| ItemCode | ManufacturerName |
+-----+-----+
| IT106    | LG               |
| IT111    | Sony             |
| IT116    | Onida            |
| IT121    | Onida            |
| IT123    | LG               |
+-----+-----+
5 rows in set (0.00 sec)
```

Q4. Display ItemName, ManufacturerName and UnitPrice of "Televisions, Mobiles and Washing Machines".

```
mysql> Select ItemName, ManufacturerName, UnitPrice from item where ItemCategory in ('Television','Mobile','WashingMachine');
+-----+-----+-----+
| ItemName      | ManufacturerName | UnitPrice |
+-----+-----+-----+
| LED32Inch     | Samsung         | 15000.00  |
| GalaxyS5      | Samsung         | 13000.00  |
| FrontLoad     | Samsung         | 20000.00  |
| LED32Inch     | LG              | 19000.00  |
| LGG5Silver    | LG              | 11000.00  |
| TopLoad       | LG              | 25000.00  |
| LED32Inch     | Sony            | 22000.00  |
| SonyXperiaZ5  | Sony            | 9000.00   |
| FullyAutomatic | Sony            | 22000.00  |
| LED32Inch     | Onida           | 20000.00  |
| Onidai505     | Onida           | 10000.00  |
| SemiAutomatic | Onida           | 11000.00  |
| LED50Inch     | Onida           | 48000.00  |
| LED50Inch     | LG              | 38000.00  |
+-----+-----+-----+
14 rows in set (0.00 sec)
```

Q5. Display ItemCode, ItemName and ManufacturerName of items which were manufactured in the year 2015.

```
mysql> Select ItemCode, ItemName, ManufacturerName from item where year(ManufacturingYear)='2015';
+-----+-----+-----+
| ItemCode | ItemName      | ManufacturerName |
+-----+-----+-----+
| IT101    | LED32Inch     | Samsung         |
| IT104    | DualDoor      | Samsung         |
| IT107    | LGG5Silver    | LG              |
| IT112    | SonyXperiaZ5  | Sony            |
| IT113    | FullyAutomatic | Sony            |
| IT118    | SemiAutomatic | Onida           |
+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> _
```

Q6. Display ItemCode, ItemName and ManufacturerName of Televisions which are more than 2 years old

```
mysql> Select ItemCode, ItemName, ManufacturerName from item where TIMESTAMPDIFF(Year, ManufacturingYear, now()) > 2;
```

ItemCode	ItemName	ManufacturerName
IT101	LED32Inch	Samsung
IT102	GalaxyS5	Samsung
IT103	FrontLoad	Samsung
IT104	DualDoor	Samsung
IT105	Pencils	Natraj
IT106	LED32Inch	LG
IT107	LGG5Silver	LG
IT108	TopLoad	LG
IT109	FourDoor	LG
IT110	Eraser	Camlin
IT111	LED32Inch	Sony
IT112	SonyXperiaZ5	Sony
IT113	FullyAutomatic	Sony
IT114	ThreeDoorNormal	Sony
IT115	Pens	FaberCastell
IT116	LED32Inch	Onida
IT117	Onidai505	Onida
IT118	SemiAutomatic	Onida
IT119	ThreeDoorLux	Onida
IT120	Sharpner	Apsara
IT121	LED50Inch	Onida
IT122	ThreeDoor	Thomson
IT123	LED50Inch	LG

```
23 rows in set (0.00 sec)
```

Q7. 7. Display ItemCode, ItemName, UnitPrice, UnitPrice+VAT for all "Refrigerators" (VAT to be computed as 2% of unitprice of the item).

```
mysql> Select ItemCode,ItemName, UnitPrice, UnitPrice+(2/100)*UnitPrice AS UnitPriceVAT from item;
```

ItemCode	ItemName	UnitPrice	UnitPriceVAT
IT101	LED32Inch	15000.00	15300.000000
IT102	GalaxyS5	13000.00	13260.000000
IT103	FrontLoad	20000.00	20400.000000
IT104	DualDoor	12000.00	12240.000000
IT105	Pencils	5.00	5.100000
IT106	LED32Inch	19000.00	19380.000000
IT107	LGG5Silver	11000.00	11220.000000
IT108	TopLoad	25000.00	25500.000000
IT109	FourDoor	18000.00	18360.000000
IT110	Eraser	10.00	10.200000
IT111	LED32Inch	22000.00	22440.000000
IT112	SonyXperiaZ5	9000.00	9180.000000
IT113	FullyAutomatic	22000.00	22440.000000
IT114	ThreeDoorNormal	19000.00	19380.000000
IT115	Pens	5.00	5.100000
IT116	LED32Inch	20000.00	20400.000000
IT117	Onidai505	10000.00	10200.000000
IT118	SemiAutomatic	11000.00	11220.000000
IT119	ThreeDoorLux	18000.00	18360.000000
IT120	Sharpner	10.00	10.200000
IT121	LED50Inch	48000.00	48960.000000
IT122	ThreeDoor	21000.00	21420.000000
IT123	LED50Inch	38000.00	38760.000000
IT124	ThreeDoor	22000.00	22440.000000
IT125	FourDoor	21000.00	21420.000000

```
25 rows in set (0.00 sec)
```

Q8. Display CustomerName and PhoneNumber of Customers whose name starts with 'J'.

For the Right Syntax to use near like use like 'J%';

```
mysql> Select CustomerName, PhoneNumber from Customer where CustomerName like 'J%';
```

CustomerName	PhoneNumber
Jacob	1234567895
James	1234567896

2 rows in set (0.00 sec)

Q9. Display CustomerName and Phone of Customers whose name ends with 'an'.

```
mysql> Select CustomerName, PhoneNumber from Customer where CustomerName like '%an';
```

CustomerName	PhoneNumber
Megan	1234567891
Dan	1234567897
Morgan	1234567903

3 rows in set (0.00 sec)

Q10. Display CustomerName and Phone of Customers whose name starts with 'M' and ends with 'n'.

```
mysql> Select CustomerName, PhoneNumber from Customer where CustomerName like 'M%n';
```

CustomerName	PhoneNumber
Megan	1234567891
Mecon	1234567901
Merlin	1234567902
Morgan	1234567903

4 rows in set (0.00 sec)

Q11. Display CustomerName and Phone of Customers whose name contains only three letters.

```
mysql> Select CustomerName, PhoneNumber from Customer where CustomerName like '___';
```

CustomerName	PhoneNumber
Amy	1234567892
Dan	1234567897
Ken	1234567900

3 rows in set (0.00 sec)

Q12. Display CustomerName and Phone of Customers whose name contains 'a' as the second letter.

```
mysql> Select CustomerName, PhoneNumber from Customer where CustomerName like '_a%';
```

CustomerName	PhoneNumber
Mario	1234567890
Jacob	1234567895
James	1234567896
Dan	1234567897

```
4 rows in set (0.00 sec)
```

Q13. Display ItemCode, ItemName, ManufacturerName and UnitPrice of ALL items in the descending order of UnitPrice.

```
mysql> Select ItemCode,ItemName, ManufacturerName, UnitPrice from item order by UnitPrice desc;
```

ItemCode	ItemName	ManufacturerName	UnitPrice
IT121	LED50Inch	Onida	48000.00
IT123	LED50Inch	LG	38000.00
IT108	TopLoad	LG	25000.00
IT113	FullyAutomatic	Sony	22000.00
IT124	ThreeDoor	Samsung	22000.00
IT111	LED32Inch	Sony	22000.00
IT122	ThreeDoor	Thomson	21000.00
IT125	FourDoor	Samsung	21000.00
IT103	FrontLoad	Samsung	20000.00
IT116	LED32Inch	Onida	20000.00
IT106	LED32Inch	LG	19000.00
IT114	ThreeDoorNormal	Sony	19000.00
IT109	FourDoor	LG	18000.00
IT119	ThreeDoorLux	Onida	18000.00
IT101	LED32Inch	Samsung	15000.00
IT102	GalaxyS5	Samsung	13000.00
IT104	DualDoor	Samsung	12000.00
IT118	SemiAutomatic	Onida	11000.00
IT107	LGG5Silver	LG	11000.00
IT117	Onidai505	Onida	10000.00
IT112	SonyXperiaZ5	Sony	9000.00
IT120	Sharpner	Apsara	10.00
IT110	Eraser	Camlin	10.00
IT105	Pencils	Natraj	5.00
IT115	Pens	FaberCastell	5.00

```
25 rows in set (0.00 sec)
```

Q14. Display CustomerName and Address of customers in the alphabetical order (A to Z) of CustomerName.

```
mysql> Select CustomerName, Address from customer order by customerName;
```

CustomerName	Address
Amy	Street: 1, Cross: 2, Town: 3, Pin: 1231
Dan	Street: 2, Cross: 4, Town: 5, Pin: 1232
Eric	Street: 12, Cross: 4, Town: 15, Pin: 1235
Henry	Street: 12, Cross: 4, Town: 15, Pin: 1235
Jacob	Street: 2, Cross: 4, Town: 5, Pin: 1232
James	Street: 2, Cross: 4, Town: 5, Pin: 1232
Ken	Street: 12, Cross: 4, Town: 15, Pin: 1235
Mario	Street: 1, Cross: 2, Town: 3, Pin: 1231
Mecon	Street: 14, Cross: 6, Town: 16, Pin: 1236
Megan	Street: 1, Cross: 2, Town: 3, Pin: 1231
Merlin	Street: 14, Cross: 5, Town: 16, Pin: 1236
Morgan	Street: 14, Cross: 5, Town: 16, Pin: 1236
Phil	Street: 2, Cross: 4, Town: 5, Pin: 1232
Stuart	Street: 1, Cross: 2, Town: 3, Pin: 1231

```
14 rows in set (0.00 sec)
```

Q15. Display ItemCode, ItemName, ManufacturerName and UnitPrice of all items in the ascending order of UnitPrice and in the descending order of ManufacturerName (if UnitPrice is same).

```
mysql> Select ItemCode,ItemName, ManufacturerName, UnitPrice from item order by UnitPrice, ManufacturerName desc;
```

ItemCode	ItemName	ManufacturerName	UnitPrice
IT105	Pencils	Natraj	5.00
IT115	Pens	FaberCastell	5.00
IT110	Eraser	Camlin	10.00
IT120	Sharpner	Apsara	10.00
IT112	SonyXperiaZ5	Sony	9000.00
IT117	OnidaI505	Onida	10000.00
IT118	SemiAutomatic	Onida	11000.00
IT107	LGG5Silver	LG	11000.00
IT104	DualDoor	Samsung	12000.00
IT102	GalaxyS5	Samsung	13000.00
IT101	LED32Inch	Samsung	15000.00
IT119	ThreeDoorLux	Onida	18000.00
IT109	FourDoor	LG	18000.00
IT114	ThreeDoorNormal	Sony	19000.00
IT106	LED32Inch	LG	19000.00
IT103	FrontLoad	Samsung	20000.00
IT116	LED32Inch	Onida	20000.00
IT122	ThreeDoor	Thomson	21000.00
IT125	FourDoor	Samsung	21000.00
IT111	LED32Inch	Sony	22000.00
IT113	FullyAutomatic	Sony	22000.00
IT124	ThreeDoor	Samsung	22000.00
IT108	TopLoad	LG	25000.00
IT123	LED50Inch	LG	38000.00
IT121	LED50Inch	Onida	48000.00

```
25 rows in set (0.00 sec)
```

Q1.Increase the salary of ALL employees by 5%. Save the changes done to the database table

Alter table EmployeeD Add increasedSalary numeric(10,2);

```
mysql> Update EmployeeD set increasedSalary = Salary +(5/100)*Salary;
Query OK, 19 rows affected (0.00 sec)
Rows matched: 19  Changed: 19  Warnings: 0

mysql> Select * from EmployeeD;
```

EmployeeNumber	EmployeeName	DateOfJoining	DateOfBirth	Designation	Salary	ManagerEmployeeNumber	DepartmentCode	increasedSalary
7001	Cynthia	1975-05-12	1997-02-14	CEO	800000.00	7001	NULL	840000.00
7002	Mario	1976-02-14	1998-04-16	MD	500000.00	7001	JavaCap	525000.00
7003	Jacob	1976-05-16	1998-05-16	MD	400000.00	7001	.NETCap	420000.00
7004	Lucy	1978-05-15	2000-07-15	MD	420000.00	7001	LKM	441000.00
7005	Amy	1978-09-16	2000-11-16	SM	240000.00	7002	JavaCap	252000.00
7006	Frank	1978-09-17	2000-09-19	SM	220000.00	7003	.NETCap	231000.00
7007	Phil	1974-12-11	2000-11-12	SM	220000.00	7004	LKM	231000.00
7008	Arnold	1984-03-13	2000-04-01	TL	80000.00	7005	JavaCap	84000.00
7009	Jack	1984-09-23	2000-06-23	TL	88000.00	7006	.NETCap	92400.00
7010	Justin	1984-11-07	2000-02-09	TL	86000.00	7007	LKM	90300.00
7011	Megan	1984-07-21	2002-09-19	TL	87000.00	7007	LKM	91350.00
7012	Stuart	1980-05-23	2016-05-22	SSE	35000.00	7008	JavaCap	36750.00
7013	Clarke	1994-02-24	2016-05-22	SSE	32000.00	7008	JavaCap	33600.00
7014	Darwin	1992-05-03	2016-05-22	SE	30000.00	7009	.NETCap	31500.00
7015	Chelsea	1994-01-19	2016-05-22	SSE	38000.00	7010	LKM	39900.00
7016	Dan	1991-05-27	2016-07-07	SE	30000.00	7009	.NETCap	31500.00
7017	Jimmy	1993-08-11	2016-07-07	SE	32000.00	7010	LKM	33600.00
7018	James	1993-12-19	2016-07-07	SE	35000.00	NULL	.NETCap	36750.00
7019	Joseph	1992-12-31	2016-07-07	SE	30000.00	NULL	.NETCap	31500.00

19 rows in set (0.00 sec)

Q2.Increase the salary of SSEs by 5% in addition to increase done in the previous statement. Save the changes done to the database table

```
mysql> Update EmployeeD set increasedSalary = increasedSalary +(5/100)*increasedSalary;
Query OK, 19 rows affected (0.01 sec)
Rows matched: 19  Changed: 19  Warnings: 0

mysql> Select * from EmployeeD;
```

EmployeeNumber	EmployeeName	DateOfJoining	DateOfBirth	Designation	Salary	ManagerEmployeeNumber	DepartmentCode	increasedSalary
7001	Cynthia	1975-05-12	1997-02-14	CEO	800000.00	7001	NULL	882000.00
7002	Mario	1976-02-14	1998-04-16	MD	500000.00	7001	JavaCap	551250.00
7003	Jacob	1976-05-16	1998-05-16	MD	400000.00	7001	.NETCap	441000.00
7004	Lucy	1978-05-15	2000-07-15	MD	420000.00	7001	LKM	463050.00
7005	Amy	1978-09-16	2000-11-16	SM	240000.00	7002	JavaCap	264600.00
7006	Frank	1978-09-17	2000-09-19	SM	220000.00	7003	.NETCap	242550.00
7007	Phil	1974-12-11	2000-11-12	SM	220000.00	7004	LKM	242550.00
7008	Arnold	1984-03-13	2000-04-01	TL	80000.00	7005	JavaCap	88200.00
7009	Jack	1984-09-23	2000-06-23	TL	88000.00	7006	.NETCap	97020.00
7010	Justin	1984-11-07	2000-02-09	TL	86000.00	7007	LKM	94815.00
7011	Megan	1984-07-21	2002-09-19	TL	87000.00	7007	LKM	95917.50
7012	Stuart	1980-05-23	2016-05-22	SSE	35000.00	7008	JavaCap	38587.50
7013	Clarke	1994-02-24	2016-05-22	SSE	32000.00	7008	JavaCap	35280.00
7014	Darwin	1992-05-03	2016-05-22	SE	30000.00	7009	.NETCap	33075.00
7015	Chelsea	1994-01-19	2016-05-22	SSE	38000.00	7010	LKM	41895.00
7016	Dan	1991-05-27	2016-07-07	SE	30000.00	7009	.NETCap	33075.00
7017	Jimmy	1993-08-11	2016-07-07	SE	32000.00	7010	LKM	35280.00
7018	James	1993-12-19	2016-07-07	SE	35000.00	NULL	.NETCap	38587.50
7019	Joseph	1992-12-31	2016-07-07	SE	30000.00	NULL	.NETCap	33075.00

19 rows in set (0.00 sec)

Q3.Delete ALL rows from "EmployeeProject" table. Undo the changes done to the database table.

```
mysql> start transaction;
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> Select * from EmployeeProjects;
```

EmployeeNumber	ProjectId	StartDate	EndDate
7004	P2	2014-07-16	2015-05-11
7005	P1	2014-07-01	NULL
7006	P1	2016-06-01	NULL
7007	P3	2015-05-11	NULL
7012	P1	2015-03-01	NULL
7012	P2	2016-06-01	2015-02-28
7013	P2	2014-07-01	2014-11-11
7013	P3	2015-02-28	NULL
7014	P3	2014-11-11	NULL
7016	P2	2014-07-16	NULL

```
10 rows in set (0.00 sec)
```

```
mysql> savepoint sp_1;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> savepoint sp_1;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> delete from EmployeeProjects;
Query OK, 10 rows affected (0.00 sec)
```

```
mysql> Select * from EmployeeProjects;
Empty set (0.00 sec)
```

```
mysql> rollback to savepoint sp_1;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> Select * from EmployeeProjects;
Empty set (0.00 sec)
```

```
mysql> rollback to savepoint sp_1;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> Select * from EmployeeProjects;
```

EmployeeNumber	ProjectId	StartDate	EndDate
7004	P2	2014-07-16	2015-05-11
7005	P1	2014-07-01	NULL
7006	P1	2016-06-01	NULL
7007	P3	2015-05-11	NULL
7012	P1	2015-03-01	NULL
7012	P2	2016-06-01	2015-02-28
7013	P2	2014-07-01	2014-11-11
7013	P3	2015-02-28	NULL
7014	P3	2014-11-11	NULL
7016	P2	2014-07-16	NULL

```
10 rows in set (0.00 sec)
```


Q4.Delete rows from "EmployeeProject" table if the employee is working for project 'P1'. Undo the changes done to the database table

```
mysql> start transaction;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from EmployeeProjects;
+-----+-----+-----+-----+
| EmployeeNumber | ProjectId | StartDate | EndDate |
+-----+-----+-----+-----+
| 7004 | P2 | 2014-07-16 | 2015-05-11 |
| 7005 | P1 | 2014-07-01 | NULL |
| 7006 | P1 | 2016-06-01 | NULL |
| 7007 | P3 | 2015-05-11 | NULL |
| 7012 | P1 | 2015-03-01 | NULL |
| 7012 | P2 | 2016-06-01 | 2015-02-28 |
| 7013 | P2 | 2014-07-01 | 2014-11-11 |
| 7013 | P3 | 2015-02-28 | NULL |
| 7014 | P3 | 2014-11-11 | NULL |
| 7016 | P2 | 2014-07-16 | NULL |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> savepoint sp_1;
Query OK, 0 rows affected (0.00 sec)

mysql> delete from employeeProject where projectId='P1';
ERROR 1146 (42S02): Table 'training.employeeproject' doesn't exist
mysql> delete from employeeProjects where projectId='P1';
Query OK, 3 rows affected (0.00 sec)
```

```
mysql> select * from EmployeeProjects;
+-----+-----+-----+-----+
| EmployeeNumber | ProjectId | StartDate | EndDate |
+-----+-----+-----+-----+
| 7004 | P2 | 2014-07-16 | 2015-05-11 |
| 7007 | P3 | 2015-05-11 | NULL |
| 7012 | P2 | 2016-06-01 | 2015-02-28 |
| 7013 | P2 | 2014-07-01 | 2014-11-11 |
| 7013 | P3 | 2015-02-28 | NULL |
| 7014 | P3 | 2014-11-11 | NULL |
| 7016 | P2 | 2014-07-16 | NULL |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> rollback to savepoint sp_1;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from EmployeeProjects;
+-----+-----+-----+-----+
| EmployeeNumber | ProjectId | StartDate | EndDate |
+-----+-----+-----+-----+
| 7004 | P2 | 2014-07-16 | 2015-05-11 |
| 7005 | P1 | 2014-07-01 | NULL |
| 7006 | P1 | 2016-06-01 | NULL |
| 7007 | P3 | 2015-05-11 | NULL |
| 7012 | P1 | 2015-03-01 | NULL |
| 7012 | P2 | 2016-06-01 | 2015-02-28 |
| 7013 | P2 | 2014-07-01 | 2014-11-11 |
| 7013 | P3 | 2015-02-28 | NULL |
| 7014 | P3 | 2014-11-11 | NULL |
| 7016 | P2 | 2014-07-16 | NULL |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Q5.Delete ALL rows from "Department" table.

Delete from Department;

Q1. Display EmployeeName, Designation and Salary for ALL the employees

```
mysql> Select EmployeeName, Designation, Salary from EmployeeD;
```

EmployeeName	Designation	Salary
Cynthia	CEO	800000.00
Mario	MD	500000.00
Jacob	MD	400000.00
Lucy	MD	420000.00
Amy	SM	240000.00
Frank	SM	220000.00
Phil	SM	220000.00
Arnold	TL	80000.00
Jack	TL	88000.00
Justin	TL	86000.00
Megan	TL	87000.00
Stuart	SSE	35000.00
Clarke	SSE	32000.00
Darwin	SE	30000.00
Chelsea	SSE	38000.00
Dan	SE	30000.00
Jimmy	SE	32000.00
James	SE	35000.00
Joseph	SE	30000.00

```
19 rows in set (0.00 sec)
```

Q2.Display different designations in the company (Each designation should be displayed only once)

```
mysql> Select distinct designation from employeeD;
```

designation
CEO
MD
SM
TL
SSE
SE

```
6 rows in set (0.00 sec)
```

Q3.Display EmployeeName and Salary of SSEs whose salary is more than 35000

```
mysql> Select EmployeeName, Salary from EmployeeD where salary > 35000 And Designation = 'SSE';
```

EmployeeName	Salary
Chelsea	38000.00

```
1 row in set (0.00 sec)
```

Q4.Display EmployeeName, Designation and Salary of SM, SSE and SE

```
mysql> Select EmployeeName,salary, Designation from EmployeeD where Designation in('SM', 'SSE', 'SE');
```

EmployeeName	salary	Designation
Amy	240000.00	SM
Frank	220000.00	SM
Phil	220000.00	SM
Stuart	35000.00	SSE
Clarke	32000.00	SSE
Darwin	30000.00	SE
Chelsea	38000.00	SSE
Dan	30000.00	SE
Jimmy	32000.00	SE
James	35000.00	SE
Joseph	30000.00	SE

```
11 rows in set (0.00 sec)
```

Q5.Display EmployeeName and DateOfJoining of employees who have joined in the year 1976.

```
mysql> Select EmployeeName, DateOfJoining from EmployeeD where year(DateOfJoining) = 1976;
```

EmployeeName	DateOfJoining
Mario	1976-02-14
Jacob	1976-05-16

```
2 rows in set (0.00 sec)
```

Q6.Display EmployeeName, DateofBirth and Age of ALL employees (Age is not a database column.

Needs to be computed. In Oracle, SYSDATE contains the current date)

```
mysql> Select employeeName, DateOfBirth, Timestampdiff(Year, DateOfBirth, now()) as Age from EmployeeD;
```

employeeName	DateOfBirth	Age
Cynthia	1997-02-14	25
Mario	1998-04-16	23
Jacob	1998-05-16	23
Lucy	2000-07-15	21
Amy	2000-11-16	21
Frank	2000-09-19	21
Phil	2000-11-12	21
Arnold	2000-04-01	21
Jack	2000-06-23	21
Justin	2000-02-09	22
Megan	2002-09-19	19
Stuart	2016-05-22	5
Clarke	2016-05-22	5
Darwin	2016-05-22	5
Chelsea	2016-05-22	5
Dan	2016-07-07	5
Jimmy	2016-07-07	5
James	2016-07-07	5
Joseph	2016-07-07	5

```
19 rows in set (0.00 sec)
```

Q7.Display EmployeeName and Salary of employees whose salary is in the range INR 50,000 to INR 100,000

```
mysql> Select EmployeeName, Salary from employeeD where salary between 50000 And 100000;
```

EmployeeName	Salary
Arnold	80000.00
Jack	88000.00
Justin	86000.00
Megan	87000.00

```
4 rows in set (0.00 sec)
```

8.Display EmployeeName of employees whose name starts with 'J'

```
mysql> Select EmployeeName from EmployeeD where EmployeeName like 'J%';
+-----+
| EmployeeName |
+-----+
| Jacob        |
| Jack         |
| Justin       |
| Jimmy        |
| James        |
| Joseph       |
+-----+
6 rows in set (0.00 sec)
```

9.Display EmployeeName of employees whose name ends with 'k'

```
mysql> Select EmployeeName from EmployeeD where EmployeeName like '%k';
+-----+
| EmployeeName |
+-----+
| Frank        |
| Jack         |
+-----+
2 rows in set (0.00 sec)
```

10.Display EmployeeName of employees whose names contains 'a' as the second letter

```
mysql> Select EmployeeName from EmployeeD where EmployeeName like '_a%';
+-----+
| EmployeeName |
+-----+
| Mario        |
| Jacob        |
| Jack         |
| Darwin       |
| Dan          |
| James        |
+-----+
6 rows in set (0.00 sec)
```

11.Display EmployeeName of employees whose names contains only three letters

```
mysql> Select EmployeeName from EmployeeD where EmployeeName like '____';
```

EmployeeName
Amy
Dan

```
2 rows in set (0.00 sec)
```

12.Display EmployeeName and Designation of MDs whose name starts with 'M'

```
mysql> Select EmployeeName, Designation from EmployeeD where Designation='MD' And Employee
Name like 'M%';
```

EmployeeName	Designation
Mario	MD

```
1 row in set (0.00 sec)
```

13.Display EmployeeName and DateOfJoining of employees who have joined in the month of 'MAY'

```
mysql> Select EmployeeName, DateOfJoining from employeeD where month(DateOfJoining)=05;
```

EmployeeName	DateOfJoining
Cynthia	1975-05-12
Jacob	1976-05-16
Lucy	1978-05-15
Stuart	1980-05-23
Darwin	1992-05-03
Dan	1991-05-27

```
6 rows in set (0.00 sec)
```

1.Display EmployeeName and Salary of ALL employees in the alphabetical order ('A' to 'z') of

EmployeeName

```
mysql> Select EmployeeName, Salary from EmployeeD order by EmployeeName;
```

EmployeeName	Salary
Amy	240000.00
Arnold	80000.00
Chelsea	38000.00
Clarke	32000.00
Cynthia	800000.00
Dan	30000.00
Darwin	30000.00
Frank	220000.00
Jack	88000.00
Jacob	400000.00
James	35000.00
Jimmy	32000.00
Joseph	30000.00
Justin	86000.00
Lucy	420000.00
Mario	500000.00
Megan	87000.00
Phil	220000.00
Stuart	35000.00

```
19 rows in set (0.00 sec)
```

2.Display EmployeeName and DateOfBirth of ALL employees in the order of eldest to youngest

```
mysql> Select EmployeeName, DateOfBirth from EmployeeD order by timestampdiff(year, DateOfBirth, now()) desc;
```

EmployeeName	DateOfBirth
Cynthia	1997-02-14
Mario	1998-04-16
Jacob	1998-05-16
Justin	2000-02-09
Amy	2000-11-16
Lucy	2000-07-15
Jack	2000-06-23
Arnold	2000-04-01
Phil	2000-11-12
Frank	2000-09-19
Megan	2002-09-19
James	2016-07-07
Jimmy	2016-07-07
Dan	2016-07-07
Chelsea	2016-05-22
Darwin	2016-05-22
Clarke	2016-05-22
Stuart	2016-05-22
Joseph	2016-07-07

```
19 rows in set (0.00 sec)
```

3.Display EmployeeName and Salary of ALL employees in the decreasing order of Salary

```
mysql> Select EmployeeName, Salary from EmployeeD order by Salary desc;
```

EmployeeName	Salary
Cynthia	800000.00
Mario	500000.00
Lucy	420000.00
Jacob	400000.00
Amy	240000.00
Frank	220000.00
Phil	220000.00
Jack	88000.00
Megan	87000.00
Justin	86000.00
Arnold	80000.00
Chelsea	38000.00
Stuart	35000.00
James	35000.00
Clarke	32000.00
Jimmy	32000.00
Darwin	30000.00
Dan	30000.00
Joseph	30000.00

```
19 rows in set (0.00 sec)
```

4.Display EmployeeName and Salary of ALL employees in the decreasing order of Salary and in the alphabetical order of ('A' to 'z') EmployeeName if the salary is same

```
mysql> Select EmployeeName, Salary from EmployeeD order by Salary desc, employeeName;
```

EmployeeName	Salary
Cynthia	800000.00
Mario	500000.00
Lucy	420000.00
Jacob	400000.00
Amy	240000.00
Frank	220000.00
Phil	220000.00
Jack	88000.00
Megan	87000.00
Justin	86000.00
Arnold	80000.00
Chelsea	38000.00
James	35000.00
Stuart	35000.00
Clarke	32000.00
Jimmy	32000.00
Dan	30000.00
Darwin	30000.00
Joseph	30000.00

```
19 rows in set (0.00 sec)
```


5.Display EmployeeName,Designation and Salary of TLs in the decreasing order of Salary

```
mysql> Select EmployeeName, Designation, Salary from EmployeeD where Designation='TL' order by Salary;
```

EmployeeName	Designation	Salary
Arnold	TL	80000.00
Justin	TL	86000.00
Megan	TL	87000.00
Jack	TL	88000.00

```
4 rows in set (0.00 sec)
```

ASSIGNMENT - 2

1. Display UnitPrice which is the maximum in the store

```
mysql> Select max(UnitPrice) from item;
```

max(UnitPrice)
48000.00

```
1 row in set (0.00 sec)
```

2. Display the total number of items in the store.

//Using Group By in Select Statement

```
mysql> Select ManufacturerName,count(*) from item group by ManufacturerName;
```

ManufacturerName	count(*)
Apsara	1
Camlin	1
FaberCastell	1
LG	5
Natraj	1
Onida	5
Samsung	6
Sony	4
Thomson	1

```
9 rows in set (0.00 sec)
```

3. Display customerId and total number of orders placed by each customer.

```
mysql> select CustomerId, count(*) from Customer group by CustomerId;
```

CustomerId	count(*)
1001	1
1002	1
1003	1
1004	1
1005	1
1006	1
1007	1
1008	1
1009	1
1010	1
1011	1
1012	1
1013	1
1014	1

```
14 rows in set (0.00 sec)
```

4. Display ItemCategory and average UnitPrice in each item category.

```
mysql> Select ItemCategory, avg(UnitPrice) from Item group by ItemCategory;
```

ItemCategory	avg(UnitPrice)
Mobile	10750.000000
Refrigerator	18714.285714
Stationary	7.500000
Television	27000.000000
WashingMachine	19500.000000

```
5 rows in set (0.00 sec)
```

5. Display customerId of customers who have placed more than 1 order.

```
mysql> select CustomerId, count(OrderId) from orderMaster group by customerId having count(
(O
```

CustomerId	count(OrderId)
1005	3
1006	3

```
2 rows in set (0.00 sec)
```

6. Display ItemCategory of items which has the minimum unit price more than INR 10,000.

```
mysql> Select ItemCategory,UnitPrice from item where UnitPrice > 10000 group by ItemCategory;
+-----+-----+
| ItemCategory | UnitPrice |
+-----+-----+
| Mobile       | 13000.00  |
| Refrigerator | 12000.00  |
| Television   | 15000.00  |
| WashingMachine | 20000.00  |
+-----+-----+
4 rows in set (0.00 sec)
```

7. Display ItemCategory, total number of items for "Television" and "Refrigerator" if the total number of items exceeds 5. Display the results in the descending order of total number of items.

```
mysql> Select ItemCategory, count(*) from item group by ItemCategory having ItemCategory in('Television', 'Refrigerator') and count(*)>5 order by count(*) desc;
+-----+-----+
| ItemCategory | count(*) |
+-----+-----+
| Refrigerator | 7        |
| Television   | 6        |
+-----+-----+
2 rows in set (0.00 sec)
```

Data Retrieval Language – Subqueries and Joins

1. Display CustomerName and PhoneNumber of customers who have placed a single order of more than INR 45,000.

```
mysql> Select CustomerName, PhoneNumber from customer where customerId in(Select customerId from ordermaster where TotalOrderedAmount > 45000 group by customerId having count(CustomerId) =1);
+-----+-----+
| CustomerName | PhoneNumber |
+-----+-----+
| Mario        | 1234567890 |
+-----+-----+
1 row in set (0.00 sec)
```

2. Display CustomerName and PhoneNumber of Customers who have placed orders in the month of June in year 2016.

```
mysql> Select CustomerName, PhoneNumber from Customer where customerId in(Select customerId from ordermaster where year(
OrderDate) = 2016 and month(OrderDate) = 06);
+-----+-----+
| CustomerName | PhoneNumber |
+-----+-----+
| Jacob        | 1234567895  |
| Phil         | 1234567894  |
+-----+-----+
2 rows in set (0.00 sec)
```

3. Display ItemCode, ItemName and UnitPrice of items which were not ordered by any customer.

```
mysql> Select ItemCode,ItemName, UnitPrice from item where ItemCode not in (Select ItemCode from OrderTransaction);
+-----+-----+-----+
| ItemCode | ItemName      | UnitPrice |
+-----+-----+-----+
| IT104    | DualDoor      | 12000.00  |
| IT105    | Pencils       | 5.00      |
| IT106    | LED32Inch     | 19000.00  |
| IT107    | LGG5Silver    | 11000.00  |
| IT108    | TopLoad       | 25000.00  |
| IT110    | Eraser        | 10.00     |
| IT112    | SonyXperiaZ5  | 9000.00   |
| IT113    | FullyAutomatic | 22000.00  |
| IT116    | LED32Inch     | 20000.00  |
| IT118    | SemiAutomatic | 11000.00  |
| IT119    | ThreeDoorLux  | 18000.00  |
| IT120    | Sharpner      | 10.00     |
| IT122    | ThreeDoor     | 21000.00  |
| IT123    | LED50Inch     | 38000.00  |
| IT124    | ThreeDoor     | 22000.00  |
| IT125    | FourDoor      | 21000.00  |
+-----+-----+-----+
16 rows in set (0.00 sec)
```

4. Display itemCode and ItemName of items that are ordered exactly once.

```
mysql> Select itemCode, ItemName from item where itemCode in(Select ItemCode from orderTransaction where QtyOrdered = 1)
;
+-----+-----+
| itemCode | ItemName      |
+-----+-----+
| IT103    | FrontLoad     |
| IT109    | FourDoor      |
| IT114    | ThreeDoorNormal |
| IT121    | LED50Inch     |
+-----+-----+
4 rows in set (0.00 sec)
```

5. Display customerId of customers who have placed more than one order.

```
mysql> Select CustomerId from ordermaster group by customerId having count(customerId)>1;
+-----+
| CustomerId |
+-----+
|      1005 |
|      1006 |
+-----+
2 rows in set (0.00 sec)
```

6. Display CustomerName and PhoneNumber of Customers who have placed order for most expensive item in the store.

7. Display OrderId, CustomerId, CustomerName, Address and PhoneNumber for all the orders placed.

```
mysql> Select o.orderId,c.CustomerId, c.CustomerName, c.Address, c.PhoneNumber from customer as c inner join ordermaster
as o on o.customerId = c.customerId;
+-----+-----+-----+-----+-----+
| orderId | CustomerId | CustomerName | Address | PhoneNumber |
+-----+-----+-----+-----+-----+
| 70001 | 1001 | Mario | Street: 1, Cross: 2, Town: 3, Pin: 1231 | 1234567890 |
| 70002 | 1006 | Jacob | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567895 |
| 70003 | 1005 | Phil | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567894 |
| 70004 | 1006 | Jacob | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567895 |
| 70005 | 1005 | Phil | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567894 |
| 70006 | 1006 | Jacob | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567895 |
| 70007 | 1005 | Phil | Street: 2, Cross: 4, Town: 5, Pin: 1232 | 1234567894 |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

8. Display ItemCode, ItemName, QtyOrdered, UnitPrice for OrderId 70002.

```
mysql> Select i.ItemCode, i.ItemName, o.QtyOrdered, i.UnitPrice,o.orderId from item as i, ordertransaction as o where i.
ItemCode = o.ItemCode and i.itemCode in(Select itemCode from Ordertransaction where orderId = 70002);
+-----+-----+-----+-----+-----+
| ItemCode | ItemName | QtyOrdered | UnitPrice | orderId |
+-----+-----+-----+-----+-----+
| IT103 | FrontLoad | 1 | 20000.00 | 70002 |
| IT109 | FourDoor | 1 | 18000.00 | 70002 |
| IT111 | LED32Inch | 2 | 22000.00 | 70002 |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

9. Display OrderId, CustomerId, CustomerName, Address and Phone for all the orders placed. Include the details about the customer even if there are no orders placed by the customer.

```
mysql> Select o.orderId, c.customerId, c.Address, c.PhoneNumber from customer as c Left join ordermaster as o on c.customerId = o.customerId;
```

orderId	customerId	Address	PhoneNumber
70001	1001	Street: 1, Cross: 2, Town: 3, Pin: 1231	1234567890
70002	1006	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567895
70003	1005	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567894
70004	1006	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567895
70005	1005	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567894
70006	1006	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567895
70007	1005	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567894
NULL	1002	Street: 1, Cross: 2, Town: 3, Pin: 1231	1234567891
NULL	1003	Street: 1, Cross: 2, Town: 3, Pin: 1231	1234567892
NULL	1004	Street: 1, Cross: 2, Town: 3, Pin: 1231	1234567893
NULL	1007	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567896
NULL	1008	Street: 2, Cross: 4, Town: 5, Pin: 1232	1234567897
NULL	1009	Street: 12, Cross: 4, Town: 15, Pin: 1235	1234567898
NULL	1010	Street: 12, Cross: 4, Town: 15, Pin: 1235	1234567899
NULL	1011	Street: 12, Cross: 4, Town: 15, Pin: 1235	1234567900
NULL	1012	Street: 14, Cross: 6, Town: 16, Pin: 1236	1234567901
NULL	1013	Street: 14, Cross: 5, Town: 16, Pin: 1236	1234567902
NULL	1014	Street: 14, Cross: 5, Town: 16, Pin: 1236	1234567903

18 rows in set (0.00 sec)

10. Display the ItemCode, UnitPrice of the all the items in each ItemCategory where the unitprice is less than the average unitprice for the itemcategory.

```
mysql> Select i.ItemCode, i.UnitPrice from item i where ItemCode in (Select ItemCode from Item i1 group by ItemCategory having i1.unitPrice < avg(i1.unitPrice));
```

ItemCode	UnitPrice
IT101	15000.00
IT104	12000.00
IT105	5.00

3 rows in set (0.00 sec)

Aggregate Functions

1. Display the maximum of salary of the company.

```
mysql> Select max(Salary) from EmployeeD;
```

max(Salary)
800000.00

1 row in set (0.00 sec)

2. Display the average salary of the company.

```
mysql> Select avg(Salary) from EmployeeD;
+-----+
| avg(Salary) |
+-----+
| 179105.263158 |
+-----+
1 row in set (0.00 sec)
```

3. Display the maximum salary of employees who are TLs.

```
mysql> Select max(Salary) from EmployeeD where Designation='TL';
+-----+
| max(Salary) |
+-----+
| 88000.00 |
+-----+
1 row in set (0.00 sec)
```

4. Display the total number of employees in the company.

```
mysql> Select count(*) from EmployeeD;
+-----+
| count(*) |
+-----+
| 19 |
+-----+
1 row in set (0.00 sec)
```

5. Display the total number of Managers in the company. (If an employee is playing the role of the supervisor for any other employee then the employee is considered as Manager).

```
mysql> Select count(ManagerEmployeeNumber) from EmployeeD;
+-----+
| count(ManagerEmployeeNumber) |
+-----+
| 17 |
+-----+
1 row in set (0.00 sec)
```

6. Display the total number SMs in the company.

```
mysql> Select count(*) from EmployeeD where Designation='SM';
+-----+
| count(*) |
+-----+
|          3 |
+-----+
1 row in set (0.00 sec)
```

GROUP BY

1. Display designation and number of employees in each designation.

```
mysql> Select Designation, count(*) from EmployeeD group By Designation;
+-----+-----+
| Designation | count(*) |
+-----+-----+
| CEO         |          1 |
| MD          |          3 |
| SE          |          5 |
| SM          |          3 |
| SSE         |          3 |
| TL          |          4 |
+-----+-----+
6 rows in set (0.00 sec)
```

2. Display designation and maximum salary for each designation.

```
mysql> Select Designation, max(Salary) from EmployeeD group By Designation;
+-----+-----+
| Designation | max(Salary) |
+-----+-----+
| CEO         | 800000.00 |
| MD          | 500000.00 |
| SE          | 35000.00 |
| SM          | 240000.00 |
| SSE         | 38000.00 |
| TL          | 88000.00 |
+-----+-----+
6 rows in set (0.00 sec)
```

3. Display Designation and maximum salary for each designation. Display the results in the decreasing order of maximum salary.


```
mysql> Select Designation, max(Salary) from EmployeeD group By Designation order by Salary desc;
```

Designation	max(Salary)
CEO	800000.00
MD	500000.00
SM	240000.00
TL	88000.00
SSE	38000.00
SE	35000.00

```
6 rows in set (0.00 sec)
```

4. Display DepartmentCode and number of employees working for each department.

```
mysql> Select DepartmentCode, count(*) from EmployeeD group By DepartmentCode;
```

DepartmentCode	count(*)
NULL	1
.NETCap	7
JavaCap	5
LKM	6

```
4 rows in set (0.00 sec)
```

5. Display Designation and maximum salary for 'TL' and 'SSE'

```
mysql> Select Designation, max(Salary) from EmployeeD group by Designation having Designation in ('TL','SSE');
```

Designation	max(Salary)
SSE	38000.00
TL	88000.00

```
2 rows in set (0.00 sec)
```

6. Display ManagerEmployeeNumber and Number of employees working under the Manager (Exclude Null from ManagerEmployeeNumber column).

```
mysql> Select count(ManagerEmployeeNumber) from EmployeeD;
```

count(ManagerEmployeeNumber)
17

```
1 row in set (0.00 sec)
```

7. Display DepartmentCode and NumberOfEmployees if the department has more than 5 employees.

```

mysql> Select DepartmentCode, Count(*) from EmployeeD group by DepartmentCode having count
(*) > 5;
+-----+-----+
| DepartmentCode | Count(*) |
+-----+-----+
| .NETCap        | 7        |
| LKM            | 6        |
+-----+-----+
2 rows in set (0.00 sec)

```

8. Display DepartmentCode and average salary if the average salary of the department is more than INR150,000 (Exclude Null under DepartmentCode column)

```

mysql> Select DepartmentCode, avg(Salary) from EmployeeD group By DepartmentCode having av
g(Salary)>150000;
+-----+-----+
| DepartmentCode | avg(Salary) |
+-----+-----+
| NULL          | 800000.000000 |
| JavaCap       | 177400.000000 |
+-----+-----+
2 rows in set (0.00 sec)

```

9. Display Designation and average salary of each designation for "LKM" department if the average salary is more than INR 35,000. Display the results in the increasing order of average salary.

```

mysql> Select Designation, avg(Salary) from EmployeeD where DepartmentCode='LKM' group by
Designation having avg(Salary)>35000 order by avg(Salary);
+-----+-----+
| Designation | avg(Salary) |
+-----+-----+
| SSE        | 38000.000000 |
| TL         | 86500.000000 |
| SM         | 220000.000000 |
| MD         | 420000.000000 |
+-----+-----+
4 rows in set (0.00 sec)

```

10. Display ProjectId, number of employees working in the project. Display the results in the decreasing order of number of employees (Exclude the results if the enddate is not null).

```

mysql> Select ProjectId, count(*) from Project group by ProjectId order by count(*) desc;
+-----+-----+
| ProjectId | count(*) |
+-----+-----+
| P1        | 1        |
| P2        | 1        |
| P3        | 1        |
| P4        | 1        |
| P5        | 1        |
+-----+-----+
5 rows in set (0.00 sec)

```

Non-Correlated Subqueries

1. Display EmployeeName, Salary of employees whose salary is more than the average salary of the company.

```
mysql> Select EmployeeName, Salary from EmployeeD where Salary > (Select avg(Salary) from EmployeeD);
```

EmployeeName	Salary
Cynthia	800000.00
Mario	500000.00
Jacob	400000.00
Lucy	420000.00
Amy	240000.00
Frank	220000.00
Phil	220000.00

```
7 rows in set (0.00 sec)
```

2. Display EmployeeName, Salary of employee(s) who is getting the lowest salary in the company.

```
mysql> Select EmployeeName, Salary from EmployeeD where salary = (Select min(Salary) from EmployeeD);
```

EmployeeName	Salary
Darwin	30000.00
Dan	30000.00
Joseph	30000.00

```
3 rows in set (0.00 sec)
```

3. Display EmployeeName of employees who are working in project 'P1'.

```
mysql> Select EmployeeName from EmployeeD where EmployeeNumber in(Select EmployeeNumber from EmployeeProjects where ProjectId = 'P1');
```

EmployeeName
Amy
Frank
Stuart

```
3 rows in set (0.00 sec)
```

4. Display ProjectName of projects which has more than 2 employees (Exclude rows if end date is not null).

```
mysql> Select ProjectName from Project where ProjectId in(Select ProjectId from EmployeeProjects where EndDate is not null group by ProjectId having count(EmployeeNumber)>2);
```

ProjectName
Insurance

```
1 row in set (0.00 sec)
```

5. Display EmployeeName of Managers who have more than three team members.

```
mysql> Select m.EmployeeName as "Manager name" from EmployeeD e, EmployeeD m where e.EmployeeNumber =m.ManagerEmployeeNumber group by m.ManagerEmployeeNumber having count(m.ManagerEmployeeNumber)>3;
+-----+
| Manager name |
+-----+
| Cynthia      |
+-----+
1 row in set (0.01 sec)
```

6. Display the second maximum salary of the company.

```
mysql> Select EmployeeNumber, Max(Salary) from EmployeeD where salary <>(Select max(Salary) from EmployeeD);
+-----+-----+
| EmployeeNumber | Max(Salary) |
+-----+-----+
| 7002           | 500000.00   |
+-----+-----+
1 row in set (0.00 sec)
```

7. Display the ProjectName of projects which currently does not have any employees.

```
mysql> Select ProjectName from Project where ProjectId Not in(Select ProjectId from EmployeeProjects group by EmployeeNumber);
+-----+
| ProjectName |
+-----+
| Banking     |
| Internal Project |
+-----+
2 rows in set (0.00 sec)
```

Correlated Subqueries

1. Display EmployeeName, Salary of employees whose salary is more than the average salary of the department they belong to.

```
mysql> Select EmployeeName, Salary from EmployeeD e1 where salary >(Select avg(salary) from EmployeeD e2 where e2.DepartmentCode = e1.DepartmentCode);
+-----+-----+
| EmployeeName | Salary |
+-----+-----+
| Mario        | 500000.00 |
| Jacob        | 400000.00 |
| Lucy         | 420000.00 |
| Amy          | 240000.00 |
| Frank        | 220000.00 |
| Phil         | 220000.00 |
+-----+-----+
6 rows in set (0.00 sec)
```

2. Display EmployeeName of employees whose manager is younger than the employee.

```
mysql> Select e1.EmployeeName from EmployeeD e1 inner join EmployeeD e2 on e1.ManagerEmployeeNumber = e2.EmployeeNumber and e1.DateOfBirth < e2.DateOfBirth;
+-----+
| EmployeeName |
+-----+
| Arnold       |
| Jack         |
| Justin       |
+-----+
3 rows in set (0.00 sec)
```

Inner Join

1. Display EmployeeName, DepartmentCode and DepartmentName of ALL employees.

```
mysql> Select e.employeeName, e.DepartmentCode, d.DepartmentName from EmployeeD as e inner
      join Department as d on e.DepartmentCode = d.departmentCode;
```

employeeName	DepartmentCode	DepartmentName
Mario	JavaCap	Java Capability
Jacob	.NETCap	Dotnet Capability
Lucy	LKM	Learning and Knowledge Management
Amy	JavaCap	Java Capability
Frank	.NETCap	Dotnet Capability
Phil	LKM	Learning and Knowledge Management
Arnold	JavaCap	Java Capability
Jack	.NETCap	Dotnet Capability
Justin	LKM	Learning and Knowledge Management
Megan	LKM	Learning and Knowledge Management
Stuart	JavaCap	Java Capability
Clarke	JavaCap	Java Capability
Darwin	.NETCap	Dotnet Capability
Chelsea	LKM	Learning and Knowledge Management
Dan	.NETCap	Dotnet Capability
Jimmy	LKM	Learning and Knowledge Management
James	.NETCap	Dotnet Capability
Joseph	.NETCap	Dotnet Capability

18 rows in set (0.00 sec)

2. Display EmployeeName, ProjectName and StartDate of employees who are currently working on the project (include only if EndDate is NULL).

```
mysql> Select e.EmployeeName, p.ProjectName, ep.StartDate from EmployeeD as e inner join E
mployeeProjects as ep inner join Project as p on ep.projectId = p.projectId;
```

EmployeeName	ProjectName	StartDate
Cynthya	Insurance	2014-07-16
Cynthya	Retail	2014-07-01
Cynthya	Retail	2016-06-01
Cynthya	Resources	2015-05-11
Cynthya	Retail	2015-03-01
Cynthya	Insurance	2016-06-01
Cynthya	Insurance	2014-07-01
Cynthya	Resources	2015-02-28
Cynthya	Resources	2014-11-11
Cynthya	Insurance	2014-07-16
Mario	Insurance	2014-07-16
Mario	Retail	2014-07-01
Mario	Retail	2016-06-01
Mario	Resources	2015-05-11
Mario	Retail	2015-03-01
Mario	Insurance	2016-06-01
Mario	Insurance	2014-07-01
Mario	Resources	2015-02-28

Outer Join

1. Display EmployeeName, ProjectId of ALL employees even if an employee is not assigned to any project (Include only if EndDate is Null).

```
mysql> Select e.employeeName, ep.ProjectId from EmployeeD as e left join EmployeeProjects
as ep on e.EmployeeNumber = ep.EmployeeNumber where ep.EndDate is null;
```

employeeName	ProjectId
Amy	P1
Frank	P1
Phil	P3
Stuart	P1
Clarke	P3
Darwin	P3
Dan	P2
Cynthya	NULL
Mario	NULL
Jacob	NULL
Arnold	NULL
Jack	NULL
Justin	NULL
Megan	NULL
Chelsea	NULL
Jimmy	NULL
James	NULL
Joseph	NULL

18 rows in set (0.00 sec)

2. Display EmployeeName, ProjectId of ALL employees even if an employee not assigned to any project (Include only if EndDate is Null). Display "Not Allocated" if ProjectId is null.

```
mysql> Select e.employeeName, coalesce(ep.ProjectId, 'Not Allocated') from EmployeeD as e
left join EmployeeProjects as ep on e.EmployeeNumber = ep.EmployeeNumber where ep.EndDate
is null;
```

employeeName	coalesce(ep.ProjectId, 'Not Allocated')
Amy	P1
Frank	P1
Phil	P3
Stuart	P1
Clarke	P3
Darwin	P3
Dan	P2
Cynthya	Not Allocated
Mario	Not Allocated
Jacob	Not Allocated
Arnold	Not Allocated
Jack	Not Allocated
Justin	Not Allocated
Megan	Not Allocated
Chelsea	Not Allocated
Jimmy	Not Allocated
James	Not Allocated
Joseph	Not Allocated

18 rows in set (0.00 sec)

SELF Join

1. Display EmployeeName, DateOfBirth of Employee, ManagerName and DateOfBirth of Manager.

```
mysql> Select e.employeeName, e.DateOfBirth, m.EmployeeName as "Manager Name", m.DateOfBir
th as "Manager DOB" from EmployeeD e inner join EmployeeD m on e.ManagerEmployeeNumber = m
.EmployeeNumber;
```

employeeName	DateOfBirth	Manager Name	Manager DOB
Cynthya	1997-02-14	Cynthya	1997-02-14
Mario	1998-04-16	Cynthya	1997-02-14
Jacob	1998-05-16	Cynthya	1997-02-14
Lucy	2000-07-15	Cynthya	1997-02-14
Amy	2000-11-16	Mario	1998-04-16
Frank	2000-09-19	Jacob	1998-05-16
Phil	2000-11-12	Lucy	2000-07-15
Arnold	2000-04-01	Amy	2000-11-16
Jack	2000-06-23	Frank	2000-09-19
Justin	2000-02-09	Phil	2000-11-12
Megan	2002-09-19	Phil	2000-11-12
Stuart	2016-05-22	Arnold	2000-04-01
Clarke	2016-05-22	Arnold	2000-04-01
Darwin	2016-05-22	Jack	2000-06-23
Chelsea	2016-05-22	Justin	2000-02-09
Dan	2016-07-07	Jack	2000-06-23
Jimmy	2016-07-07	Justin	2000-02-09

17 rows in set (0.00 sec)

