

NAME :: IBRAR BABAR

Roll No. :: 19P-0104

DataBase Lab

BS(CS):: Section (4A)

Count

/* Counting the number of Theme parks in ATTRACTION */

SELECT COUNT(PARK_CODE)

FROM ATTRACTION;

```
mysql> use themepark;
Database changed
mysql> SELECT COUNT(PARK_CODE)
      -> FROM ATTRACTION;
+-----+
| COUNT(PARK_CODE) |
+-----+
|                11 |
+-----+
1 row in set (0.23 sec)
```

/* In order to find how many different Theme parks were in the ATTRACTION table,
we use **DISTINCT** keyword. */

select count(distinct(park_code))from attraction;

```
mysql> select count(distinct(park_code))from attraction;
+-----+
| count(distinct(park_code)) |
+-----+
|                3 |
+-----+
1 row in set (0.06 sec)
```

(4.1)

Write a query that displays the number of distinct employees in the HOURS table. You should label the column "Number of Employees".

/*It is going to count the common attributes of EMP_NUM from Hours. */

select count(Distinct(EMP_NUM)) "Number of Employees"

-> from hours;

```
mysql> select count(Distinct(EMP_NUM)) "Number of Employees"
-> from hours;
+-----+
| Number of Employees |
+-----+
|          5          |
+-----+
1 row in set (0.12 sec)
```

TASK 4.2

/* COUNT(*). */

/* COUNT(*) is going to calculate all the attributes including null and not null. */

SELECT COUNT(*)

FROM ATTRACTION;

```
mysql> SELECT COUNT(*)
-> FROM ATTRACTION;
+-----+
| COUNT(*) |
+-----+
|        11 |
+-----+
1 row in set (0.00 sec)
```

SELECT COUNT(ATTRACT_NAME)

FROM ATTRACTION;

```
mysql> SELECT COUNT(ATTRACT_NAME)
-> FROM ATTRACTION;
+-----+
| COUNT(ATTRACT_NAME) |
+-----+
|          10         |
+-----+
1 row in set (0.00 sec)

mysql>
```

/* MAX and MIN */

/* it is going to find the min and max ticket price from ticket table. */

```
SELECT
MIN(TICKET_PRICE),max(TICKET_PRICE)
FROM TICKET;
```

```
mysql> SELECT
-> MIN(TICKET_PRICE),max(TICKET_PRICE)
-> FROM TICKET;
+-----+-----+
| MIN(TICKET_PRICE) | max(TICKET_PRICE) |
+-----+-----+
|          10.99    |          42.10    |
+-----+-----+
1 row in set (0.22 sec)

mysql>
```

/* SUM and AVG */

```
select avg(line_price),sum(line_qty) from sales_line;
```

```
mysql> select avg(line_price),sum(line_qty) from sales_line;
+-----+-----+
| avg(line_price) | sum(line_qty) |
+-----+-----+
|      54.245161  |          61   |
+-----+-----+
1 row in set (0.00 sec)
```

/* Group by */

/* Group by is valid only when used in conjunction with one of the SQL aggregate functions, such as COUNT, MIN, MAX, AVG and SUM. It is used when we are calculating any SQL function on the bases of PARK_CODE. */

```
SELECT PARK_CODE, MIN(TICKET_PRICE),MAX(TICKET_PRICE)
FROM TICKET
GROUP BY PARK_CODE;
```

```
mysql> SELECT PARK_CODE, MIN(TICKET_PRICE),MAX(TICKET_PRICE)
-> FROM TICKET
-> GROUP BY PARK_CODE;
```

PARK_CODE	MIN(TICKET_PRICE)	MAX(TICKET_PRICE)
FR1001	18.99	34.99
SP4533	10.99	24.99
UK3452	10.99	42.10
ZA1342	12.12	28.67

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

Task 4.4

What happens if you miss out the GROUP BY clause?

then it shows us the only maximum and minimum ticket_price from ticket_price column.

```
mysql> SELECT PARK_CODE, MIN(TICKET_PRICE),MAX(TICKET_PRICE)
-> FROM TICKET;
```

PARK_CODE	MIN(TICKET_PRICE)	MAX(TICKET_PRICE)
SP4533	10.99	42.10

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

/* HAVING */

/* HAVING clause is applied to the output of a GROUP BY operation. */

```
SELECT PARK_CODE, AVG(TICKET_PRICE)
FROM TICKET
GROUP BY PARK_CODE
HAVING AVG(TICKET_PRICE) >= 24.99;
```

```
mysql> SELECT PARK_CODE, AVG(TICKET_PRICE)
-> FROM TICKET
-> GROUP BY PARK_CODE
-> HAVING AVG(TICKET_PRICE) >= 24.99;
+-----+-----+
| PARK_CODE | AVG(TICKET_PRICE) |
+-----+-----+
| FR1001    | 24.990000         |
| UK3452    | 25.196667         |
+-----+-----+
2 rows in set (0.09 sec)
```

/* Natural Join between THEMEPARK and TICKET tables */

```
SELECT THEMEPARK.PARK_CODE, PARK_NAME, TICKET_NO,
TICKET_TYPE, TICKET_PRICE
FROM THEMEPARK, TICKET
WHERE THEMEPARK.PARK_CODE = TICKET.PARK_CODE
```

```
mysql> SELECT THEMPARK.PARK_CODE, PARK_NAME, TICKET_NO,
->
-> TICKET_TYPE, TICKET_PRICE
-> FROM THEMPARK, TICKET
-> WHERE THEMPARK.PARK_CODE = TICKET.PARK_CODE
-> ;
```

PARK_CODE	PARK_NAME	TICKET_NO	TICKET_TYPE	TICKET_PRICE
SP4533	AdventurePort	11001	Adult	24.99
SP4533	AdventurePort	11002	Child	14.99
SP4533	AdventurePort	11003	Senior	10.99
FR1001	FairyLand	13001	Child	18.99
FR1001	FairyLand	13002	Adult	34.99
FR1001	FairyLand	13003	Senior	20.99
ZA1342	GoldTown	67832	Child	18.56
ZA1342	GoldTown	67833	Adult	28.67
ZA1342	GoldTown	67855	Senior	12.12
UK3452	PleasureLand	88567	Child	22.50
UK3452	PleasureLand	88568	Adult	42.10
UK3452	PleasureLand	89720	Senior	10.99

12 rows in set (0.19 sec)

Task 4.5

```
SELECT THEMPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME,
```

```
ATTRACT_CAPACITY
```

```
FROM THEMPARK, ATTRACTION
```

```
WHERE THEMPARK.PARK_CODE = ATTRACTION.PARK_CODE;
```

```
mysql> SELECT THEMEPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME,
->
-> ATTRACT_CAPACITY
-> FROM THEMEPARK, ATTRACTION
-> WHERE THEMEPARK.PARK_CODE = ATTRACTION.PARK_CODE;
```

PARK_CODE	PARK_NAME	ATTRACT_NAME	ATTRACT_CAPACITY
FR1001	FairyLand	ThunderCoaster	34
FR1001	FairyLand	SpinningTeacups	62
FR1001	FairyLand	FlightToStars	24
FR1001	FairyLand	Ant-Trap	30
FR1001	FairyLand	Carnival	120
UK3452	PleasureLand	3D-Lego_Show	200
UK3452	PleasureLand	BlackHole2	34
UK3452	PleasureLand	Pirates	42
UK3452	PleasureLand	UnderSeaWord	80
ZA1342	GoldTown	GoldRush	80
ZA1342	GoldTown	NULL	40

11 rows in set (0.00 sec)

4.3 Joining tables with an alias

```
SELECT P.PARK_CODE, PARK_NAME, TICKET_NO, TICKET_TYPE,
```

```
TICKET_PRICE
```

```
FROM THEMEPARK P, TICKET T
```

```
WHERE P.PARK_CODE =T.PARK_CODE;
```

```
mysql> SELECT P.PARK_CODE, PARK_NAME, TICKET_NO, TICKET_TYPE,
->
-> TICKET_PRICE
->
-> FROM THEMEPARK P, TICKET T
-> WHERE P.PARK_CODE =T.PARK_CODE;
```

PARK_CODE	PARK_NAME	TICKET_NO	TICKET_TYPE	TICKET_PRICE
SP4533	AdventurePort	11001	Adult	24.99
SP4533	AdventurePort	11002	Child	14.99
SP4533	AdventurePort	11003	Senior	10.99
FR1001	FairyLand	13001	Child	18.99
FR1001	FairyLand	13002	Adult	34.99
FR1001	FairyLand	13003	Senior	20.99
ZA1342	GoldTown	67832	Child	18.56
ZA1342	GoldTown	67833	Adult	28.67
ZA1342	GoldTown	67855	Senior	12.12
UK3452	PleasureLand	88567	Child	22.50
UK3452	PleasureLand	88568	Adult	42.10
UK3452	PleasureLand	89720	Senior	10.99

12 rows in set (0.00 sec)

4.5 Natural Join

The natural join returns all rows with matching values in the matching columns and eliminates duplicate columns.

```
SELECT TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY,
LINE_PRICE
FROM SALES NATURAL JOIN SALES_LINE;
```



```
mysql> SELECT TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY,
->
-> LINE_PRICE
->
-> FROM SALES NATURAL JOIN SALES_LINE;
```

TRANSACTION_NO	SALE_DATE	LINE_NO	LINE_QTY	LINE_PRICE
12781	2007-05-18	1	2	69.98
12781	2007-05-18	2	1	14.99
12782	2007-05-18	1	2	69.98
12783	2007-05-18	1	2	41.98
12784	2007-05-18	2	1	14.99
12785	2007-05-18	1	1	14.99
12785	2007-05-18	2	1	34.99
12785	2007-05-18	3	4	139.96
34534	2007-05-18	1	4	168.40
34534	2007-05-18	2	1	22.50
34534	2007-05-18	3	2	21.98
34535	2007-05-18	1	2	84.20
34536	2007-05-18	1	2	21.98
34537	2007-05-18	1	2	84.20
34537	2007-05-18	2	1	22.50
34538	2007-05-18	1	2	21.98
34539	2007-05-18	1	2	21.98
34539	2007-05-18	2	2	84.20
34540	2007-05-18	1	4	168.40
34540	2007-05-18	2	1	22.50
34540	2007-05-18	3	2	21.98
34541	2007-05-18	1	2	84.20
67589	2007-05-18	1	2	57.34
67589	2007-05-18	2	2	37.12
67590	2007-05-18	1	2	57.34
67590	2007-05-18	2	2	37.12
67591	2007-05-18	1	1	18.56
67591	2007-05-18	2	1	12.12
67592	2007-05-18	1	4	114.68
67593	2007-05-18	1	2	57.34
67593	2007-05-18	2	2	37.12

```
31 rows in set (0.02 sec)
```

4.6 Join USING

```
SELECT TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY,
```

```
LINE_PRICE
```

FROM SALES JOIN SALES_LINE USING (TRANSACTION_NO);

```
mysql> SELECT TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY,  
-> LINE_PRICE  
-> FROM SALES JOIN SALES_LINE USING (TRANSACTION_NO);
```

TRANSACTION_NO	SALE_DATE	LINE_NO	LINE_QTY	LINE_PRICE
12781	2007-05-18	1	2	69.98
12781	2007-05-18	2	1	14.99
12782	2007-05-18	1	2	69.98
12783	2007-05-18	1	2	41.98
12784	2007-05-18	2	1	14.99
12785	2007-05-18	1	1	14.99
12785	2007-05-18	2	1	34.99
12785	2007-05-18	3	4	139.96
34534	2007-05-18	1	4	168.40
34534	2007-05-18	2	1	22.50
34534	2007-05-18	3	2	21.98
34535	2007-05-18	1	2	84.20
34536	2007-05-18	1	2	21.98
34537	2007-05-18	1	2	84.20
34537	2007-05-18	2	1	22.50
34538	2007-05-18	1	2	21.98
34539	2007-05-18	1	2	21.98
34539	2007-05-18	2	2	84.20
34540	2007-05-18	1	4	168.40
34540	2007-05-18	2	1	22.50
34540	2007-05-18	3	2	21.98
34541	2007-05-18	1	2	84.20
67589	2007-05-18	1	2	57.34
67589	2007-05-18	2	2	37.12
67590	2007-05-18	1	2	57.34
67590	2007-05-18	2	2	37.12
67591	2007-05-18	1	1	18.56
67591	2007-05-18	2	1	12.12
67592	2007-05-18	1	4	114.68
67593	2007-05-18	1	2	57.34
67593	2007-05-18	2	2	37.12

31 rows in set (0.01 sec)

6.6 Join ON

SELECT SALES.TRANSACTION_NO, SALE_DATE, LINE_NO,

LINE_QTY, LINE_PRICE

FROM SALES JOIN SALES_LINE

ON SALES.TRANSACTION_NO = SALES_LINE.TRANSACTION_NO;

```
mysql> SELECT SALES.TRANSACTION_NO, SALE_DATE, LINE_NO,  
->  
-> LINE_QTY, LINE_PRICE  
-> FROM SALES JOIN SALES_LINE  
-> ON SALES.TRANSACTION_NO = SALES_LINE.TRANSACTION_NO;
```

TRANSACTION_NO	SALE_DATE	LINE_NO	LINE_QTY	LINE_PRICE
12781	2007-05-18	1	2	69.98
12781	2007-05-18	2	1	14.99
12782	2007-05-18	1	2	69.98
12783	2007-05-18	1	2	41.98
12784	2007-05-18	2	1	14.99
12785	2007-05-18	1	1	14.99
12785	2007-05-18	2	1	34.99
12785	2007-05-18	3	4	139.96
34534	2007-05-18	1	4	168.40
34534	2007-05-18	2	1	22.50
34534	2007-05-18	3	2	21.98
34535	2007-05-18	1	2	84.20
34536	2007-05-18	1	2	21.98
34537	2007-05-18	1	2	84.20
34537	2007-05-18	2	1	22.50
34538	2007-05-18	1	2	21.98
34539	2007-05-18	1	2	21.98
34539	2007-05-18	2	2	84.20
34540	2007-05-18	1	4	168.40
34540	2007-05-18	2	1	22.50
34540	2007-05-18	3	2	21.98
34541	2007-05-18	1	2	84.20
67589	2007-05-18	1	2	57.34
67589	2007-05-18	2	2	37.12
67590	2007-05-18	1	2	57.34
67590	2007-05-18	2	2	37.12
67591	2007-05-18	1	1	18.56
67591	2007-05-18	2	1	12.12
67592	2007-05-18	1	4	114.68
67593	2007-05-18	1	2	57.34
67593	2007-05-18	2	2	37.12

31 rows in set (0.00 sec)

LEFT OUTER JOIN

SELECT THEMEPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME

FROM THEMEPARK LEFT JOIN ATTRACTION ON

THEMEPARK.PARK_CODE = ATTRACTION.PARK_CODE;

```
mysql> SELECT THEMEPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME
-> FROM THEMEPARK LEFT JOIN ATTRACTION ON THEMEPARK.PARK_CODE = ATTRACTION.PARK_CODE;
```

PARK_CODE	PARK_NAME	ATTRACT_NAME
FR1001	FairyLand	ThunderCoaster
FR1001	FairyLand	SpinningTeacups
FR1001	FairyLand	FlightToStars
FR1001	FairyLand	Ant-Trap
FR1001	FairyLand	Carnival
NL1202	Efeling	NULL
SP4533	AdventurePort	NULL
SW2323	Labyrinthe	NULL
UK2622	Miniland	NULL
UK3452	PleasureLand	3D-Lego_Show
UK3452	PleasureLand	BlackHole2
UK3452	PleasureLand	Pirates
UK3452	PleasureLand	UnderSeaWord
ZA1342	GoldTown	GoldRush
ZA1342	GoldTown	NULL

15 rows in set (0.01 sec)

RIGHT OUTER JOIN

SELECT THEMEPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME

-> FROM THEMEPARK RIGHT JOIN ATTRACTION ON

->

-> THEMEPARK.PARK_CODE = ATTRACTION.PARK_CODE;

```
mysql> SELECT THEMPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME
-> FROM THEMPARK RIGHT JOIN ATTRACTION ON
->
-> THEMPARK.PARK_CODE = ATTRACTION.PARK_CODE;
```

PARK_CODE	PARK_NAME	ATTRACT_NAME
FR1001	FairyLand	ThunderCoaster
FR1001	FairyLand	SpinningTeacups
FR1001	FairyLand	FlightToStars
FR1001	FairyLand	Ant-Trap
FR1001	FairyLand	Carnival
UK3452	PleasureLand	3D-Lego_Show
UK3452	PleasureLand	BlackHole2
UK3452	PleasureLand	Pirates
UK3452	PleasureLand	UnderSeaWord
ZA1342	GoldTown	GoldRush
ZA1342	GoldTown	NULL

11 rows in set (0.00 sec)

E4.1

select round(avg(hour_rate),2) from hours;

```
mysql> select round(avg(hour_rate),2) from hours;
```

round(avg(hour_rate),2)
7.03

1 row in set (0.06 sec)

E4.2

SELECT PARK_CODE, avg(attach_age) from attraction group by park_code having
avg(attach_age)<=7.25;

```
mysql> SELECT PARK_CODE, avg(attach_age) from attraction group by park_code having avg(attach_age)<=7.25;
```

PARK_CODE	avg(attach_age)
UK3452	7.2500

1 row in set (0.02 sec)

E4.3

select emp_num,sum(hours_per_attract) from hours group by emp_num;

```
mysql> select emp_num,sum(hours_per_attract) from hours group by emp_num;
```

emp_num	sum(hours_per_attract)
100	12
101	6
102	12
104	12
105	12

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

E4.4

```
select attract_no,min(hour_rate),max(hour_rate) from hours group by attract_no;
```

```
mysql> select attract_no,min(hour_rate),max(hour_rate) from hours group by attract_no;
```

attract_no	min(hour_rate)	max(hour_rate)
10034	6.50	6.50
10078	8.50	8.50
10098	8.50	8.50
30011	7.20	7.20
30012	5.99	7.20
30044	5.99	5.99

```
6 rows in set (0.05 sec)
```

E4.5

```
select emp_num,attract_no,avg(hours_per_attract) from hours group by emp_num,attract_no having avg(hours_per_attract)>5;
```

```
mysql> select emp_num,attract_no,avg(hours_per_attract) from hours group by emp_num,attract_no having avg(hours_per_attract)>5;
```

emp_num	attract_no	avg(hours_per_attract)
100	10034	6.0000
101	10034	6.0000
104	30011	6.0000
104	30012	6.0000

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

E4.6

```
select * from employee cross join hours;
```

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

E4.7

```
select emp_fname,emp_lname,attract_no,date_worked  
from employee natural join hours;
```

```
mysql> SELECT THEMPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME  
-> FROM THEMPARK LEFT JOIN ATTRACTION ON THEMPARK.PARK_CODE = ATTRACTION.PARK_CODE;
```

PARK_CODE	PARK_NAME	ATTRACT_NAME
FR1001	FairyLand	ThunderCoaster
FR1001	FairyLand	SpinningTeacups
FR1001	FairyLand	FlightToStars
FR1001	FairyLand	Ant-Trap
FR1001	FairyLand	Carnival
NL1202	Efling	NULL
SP4533	AdventurePort	NULL
SW2323	Labyrinthe	NULL
UK2622	MiniLand	NULL
UK3452	PleasureLand	3D-Lego_Show
UK3452	PleasureLand	BlackHole2
UK3452	PleasureLand	Pirates
UK3452	PleasureLand	UnderSeaWord
ZA1342	GoldTown	GoldRush
ZA1342	GoldTown	NULL

15 rows in set (0.01 sec)

E4.8

```
select emp_fname,emp_lname,attract_no,attract_name  
from employee natural join hours natural join attraction;
```

EMP_FNAME	EMP_LNAME	ATTRACT_NO	ATTRACT_NAME
Enna	Calderdale	10034	ThunderCoaster
Enna	Calderdale	10034	ThunderCoaster
Marshall	Ricardo	10034	ThunderCoaster
Arif	Arshad	30012	Pirates
Arif	Arshad	30044	UnderSeaWord
Arif	Arshad	30044	UnderSeaWord
Enrica	Denver	30011	BlackHole2
Enrica	Denver	30012	Pirates
Mirrelle	Namowa	10078	Ant-Trap
Mirrelle	Namowa	10098	Carnival
Mirrelle	Namowa	10098	Carnival

11 rows in set (0.00 sec)

E4.9

```
SELECT TRANSACTION_NO, SUM(LINE_PRICE) Sales
```

FROM SALES_LINE

GROUP BY TRANSACTION_NO;

select park_country from themepark where park_country = "FR";

```
mysql> SELECT TRANSACTION_NO, SUM(LINE_PRICE) Sales
-> FROM SALES_LINE
-> GROUP BY TRANSACTION_NO
-> ;
```

TRANSACTION_NO	Sales
12781	84.97
12782	69.98
12783	41.98
12784	14.99
12785	189.94
34534	212.88
34535	84.20
34536	21.98
34537	106.70
34538	21.98
34539	106.18
34540	212.88
34541	84.20
67589	94.46
67590	94.46
67591	30.68
67592	114.68
67593	94.46

18 rows in set (0.00 sec)

```
mysql> select park_country from themepark where park_country = "FR";
```

park_country
FR

1 row in set (0.00 sec)

E4.10

select sale_date,line_qty from sales natural join sales_line where sale_date ="2007-05-18";

```
mysql> select sale_date,line_qty from sales natural join sales_line where sale_date ="2007-05-18";
+-----+-----+
| sale_date | line_qty |
+-----+-----+
| 2007-05-18 | 2 |
| 2007-05-18 | 1 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 1 |
| 2007-05-18 | 1 |
| 2007-05-18 | 1 |
| 2007-05-18 | 4 |
| 2007-05-18 | 4 |
| 2007-05-18 | 1 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 1 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 4 |
| 2007-05-18 | 1 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
| 2007-05-18 | 1 |
| 2007-05-18 | 1 |
| 2007-05-18 | 4 |
| 2007-05-18 | 2 |
| 2007-05-18 | 2 |
+-----+-----+
31 rows in set (0.00 sec)
```