



TatvaSoft
sculpting thoughts...

TatvaSoft

-Ahmedabad

NAME : Makwana Sanjay I.

Technology : .PHP

Assignment 2 : Retrieve data using join with where clause

Sample table1: salesman

- salesman_id
- name
- city
- commission

Sample table2: customer

- customer_id
- cust_name
- city
- grade
- salesman_id

Sample table3: orders

- ord_no
- purch_amt
- ord_date
- customer_id
- salesman_id

1. write a SQL query to find the salesperson and customer who reside in the same city.
Return Salesman, cust_name and city.

The screenshot shows the MySQL Workbench interface. At the top, the server is 'MySQL-3306' and the database is 'assignment-2'. The 'Query' tab is active, displaying the following SQL query:

```
SELECT salesman.name AS "Salesman", customer.cust_name, customer.city FROM salesman, customer WHERE salesman.city=customer.city;
```

Below the query, a message states: 'Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)'. The query results are displayed in a table:

Salesman	cust_name	city
karan	dedhara	ahmedabad
vrushank	mehul	manipur

At the bottom, the 'Query results operations' bar includes buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

2. write a SQL query to find those orders where the order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

Server: MySQL-3306 » Database: assignment-2

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers De

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ Showing rows 0 - 6 (7 total, Query took 0.0210 seconds.)

```
SELECT a.ord_no,a.purch_amt, b.cust_name,b.city FROM orders a,customer b WHERE a.customer_id=b.customer_id AND a.purch_amt BETWEEN 500 AND 2000;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

ord_no	purch_amt	cust_name	city
3	1209	susake	delhi
14	509	nanami	vadodara
19	779	dami	surat
31	509	nanami	vadodara
36	779	suguru	gaziabad
48	509	nanami	vadodara
53	779	dami	surat

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

3. write a SQL query to find the salesperson(s) and the customer(s) he represents. Return Customer Name, city, Salesman, commission

Server: MySQL-3306 » Database: assignment-2

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Designer

Show query box

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available. ⓘ

✓ Showing rows 0 - 13 (14 total, Query took 0.0006 seconds.)

```
SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Customer Name	city	Salesman	commission
naruto	delhi	raj	100
sakura	surat	satish	101
susake	delhi	chirag	100
itachi	mumbai	akhil	10
dedhara	ahmedabad	satish	101
shikimaru	vadodara	chirag	100
joraya	delhi	pratik	587
konahamaru	surat	satish	101
gojo	mumbai	donda	56
saratu	delhi	rakesh	95
suguru	gaziabad	akhil	10
deisuke	delhi	pratik	587
hunter	gaziabad	rakesh	95
Console	delhi	smith	250

4. write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission.

Server: MySQL:3306 Database: assignment-2

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Designer

Showing rows 0 - 13 (14 total, Query took 0.0006 seconds)

```
SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE b.commission>.12;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all Number of rows: 25 Filter rows: Search this table

Extra options

Customer Name	city	Salesman	commission
naruto	delhi	raj	100
sakura	surat	satish	101
susake	delhi	chirag	100
itachi	mumbai	akhil	10
dedhara	ahmedabad	satish	101
shikimaru	vadodara	chirag	100
joraya	delhi	pratik	587
konahamaru	surat	satish	101
gojo	mumbai	donda	56
saratu	delhi	rakesh	95
suguru	gaziabad	akhil	10
deisuke	delhi	pratik	587
hunter	gaziabad	rakesh	95
rok	delhi	smith	250

☐ Show all Number of rows: 25 Filter rows: Search this table

5. write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission

Showing rows 0 - 13 (14 total, Query took 0.0007 seconds)

```
SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.city, b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE b.commission>.12 AND a.city<>b.city;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all Number of rows: 25 Filter rows: Search this table

Extra options

Customer Name	city	Salesman	city	commission
naruto	delhi	raj	vadodra	100
sakura	surat	satish	dilhi	101
susake	delhi	chirag	sura	100
itachi	mumbai	akhil	vadodra	10
dedhara	ahmedabad	satish	dilhi	101
shikimaru	vadodara	chirag	sura	100
joraya	delhi	pratik	rajsthan	587
konahamaru	surat	satish	dilhi	101
gojo	mumbai	donda	rajkot	56
saratu	delhi	rakesh	bhalar	95
suguru	gaziabad	akhil	vadodra	10
deisuke	delhi	pratik	rajsthan	587
hunter	gaziabad	rakesh	bhalar	95
rok	delhi	smith	patan	250

6. write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission

```
SELECT a.ord_no,a.ord_date,a.purch_amt, b.cust_name AS "Customer Name", b.grade, c.name AS "Salesman", c.commission FROM orders a INNER JOIN customer b ON a.customer_id=b.customer_id INNER JOIN salesman c ON a.salesman_id=c.salesman_id;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 > >> ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

ord_no	ord_date	purch_amt	Customer Name	grade	Salesman	commission
1	2023-01-12	209	naruto	A	raj	100
2	2022-10-12	29	sakura	B	satish	101
3	2022-11-12	1209	susake	C	chirag	100
4	2020-02-22	2039	itachi	A	akhil	10
5	2022-03-12	2029	dedhara	B	karan	50
6	2023-04-13	2049	shikimaru	C	donda	56
7	2022-05-16	2509	pain	A	smith	250
10	2022-08-22	2809	gojo	A	pratik	587
11	2022-09-27	2709	saratu	B	rakesh	95
13	2021-12-17	9	suguru	A	raj	100
14	2021-11-02	509	nanami	B	satish	101
15	2021-07-15	279	kambe	C	chirag	100
16	2020-06-18	249	deisuke	A	akhil	10
17	2022-05-19	239	hunter	B	karan	50
18	2012-04-15	219	ranke	C	donda	56
19	2020-02-14	779	dami	A	smith	250
21	2020-02-22	2039	naruto	A	akhil	10
■ Console	022-03-12	2029	sakura	B	karan	50

7. Write a SQL statement to join the tables salesman, customer and orders so that the same column of each table appears once and only the relational rows are returned.

[Show query box](#)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

```
SELECT * FROM orders NATURAL JOIN customer NATURAL JOIN salesman;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

salesman_id	city	customer_id	ord_no	purch_amt	ord_date	cust_name	grade	Name	Commission
-------------	------	-------------	--------	-----------	----------	-----------	-------	------	------------

Query results operations

[Create view](#)

8. write a SQL query to display the customer name, customer city, grade, salesman, salesman city. The results should be sorted by ascending customer_id.

Showing rows 0 - 22 (23 total. Query took 0.0010 seconds.)

```
SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman",b.city FROM customer a LEFT JOIN salesman b ON a.salesman_id=b.salesman_id order by a.customer_id;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

cust_name	city	grade	Salesman	city
naruto	delhi	A	raj	vadodra
sakura	surat	B	satish	dilhi
susake	delhi	C	chirag	sura
itachi	mumbai	A	akhil	vadodra
dedhara	ahmedabad	B	satish	dilhi
shikimaru	vadodara	C	chirag	sura
pain	surat	A	NULL	NULL
joraya	delhi	B	pratik	rajsthan
konahamaru	surat	C	satish	dilhi
gojo	mumbai	A	donda	rajkot
saratu	delhi	B	rakesh	bhalar
geto	vadodara	C	NULL	NULL
suguru	gaziabad	A	akhil	vadodra
nanami	vadodara	B	NULL	NULL
kambe	mumbai	C	NULL	NULL
Console	delhi	A	pratik	rajsthan

9. write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id.

Showing rows 0 - 22 (23 total. Query took 0.0010 seconds.)

```
SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", b.city FROM customer a LEFT OUTER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE a.grade<300 ORDER BY a.customer_id;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

cust_name	city	grade	Salesman	city
naruto	delhi	A	raj	vadodra
sakura	surat	B	satish	dilhi
susake	delhi	C	chirag	sura
itachi	mumbai	A	akhil	vadodra
dedhara	ahmedabad	B	satish	dilhi
shikimaru	vadodara	C	chirag	sura
pain	surat	A	NULL	NULL
joraya	delhi	B	pratik	rajsthan
konahamaru	surat	C	satish	dilhi
gojo	mumbai	A	donda	rajkot
saratu	delhi	B	rakesh	bhalar
geto	vadodara	C	NULL	NULL
suguru	gaziabad	A	akhil	vadodra
nanami	vadodara	B	NULL	NULL
kambe	mumbai	C	NULL	NULL
deisuke	delhi	A	pratik	rajsthan
hunter	gaziabad	B	rakesh	bhalar
Console	gaziabad	C	NULL	NULL

10. Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not

Showing rows 0 - 24 (56 total, Query took 0.0009 seconds.) [ord_date: ... - 2021-12-17...]

```
SELECT a.cust_name,a.city, b.ord_no, b.ord_date,b.purch_amt AS "Order Amount" FROM customer a LEFT OUTER JOIN orders b ON a.customer_id=b.customer_id order by b.ord_date;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 > >> ☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

cust_name	city	ord_no	ord_date	Order Amount
mehul	manipur	NULL	NULL	NULL
manish	mahuva	NULL	NULL	NULL
rancho	vadas	NULL	NULL	NULL
ranke	gaziabad	18	2012-04-15	219
ranke	gaziabad	52	2012-04-15	219
ranke	gaziabad	35	2012-04-15	219
suguru	gaziabad	36	2020-02-14	779
dami	surat	19	2020-02-14	779
dami	surat	53	2020-02-14	779
naruto	delhi	21	2020-02-22	2039
itachi	mumbai	38	2020-02-22	2039
itachi	mumbai	4	2020-02-22	2039
deisuke	delhi	33	2020-06-18	249
deisuke	delhi	16	2020-06-18	249
deisuke	delhi	50	2020-06-18	249
dedhara	ahmedabad	25	2021-06-17	2609
lorava	delhi	42	2021-06-17	2609

Console

11. Write a SQL statement to generate a report with customer name, city, order number, order date, order amount, salesperson name, and commission to determine if any of the existing customers have not placed orders or if they have placed orders through their salesman or by themselves.

```
SELECT a.cust_name,a.city, b.ord_no, b.ord_date,b.purch_amt AS "Order Amount", c.name,c.commission FROM customer a LEFT OUTER JOIN orders b ON a.customer_id=b.customer_id LEFT OUTER JOIN salesman c ON c.salesman_id=b.salesman_id;
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 > >> ☐ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

cust_name	city	ord_no	ord_date	Order Amount	name	commission
naruto	delhi	21	2020-02-22	2039	akhil	10
naruto	delhi	1	2023-01-12	209	raj	100
sakura	surat	22	2022-03-12	2029	karan	50
sakura	surat	2	2022-10-12	29	satish	101
susake	delhi	23	2023-04-13	2049	donda	56
susake	delhi	3	2022-11-12	1209	chirag	100
itachi	mumbai	38	2020-02-22	2039	akhil	10
itachi	mumbai	24	2022-05-16	2509	smith	250
itachi	mumbai	4	2020-02-22	2039	akhil	10
dedhara	ahmedabad	39	2022-03-12	2029	karan	50
dedhara	ahmedabad	25	2021-06-17	2609	NULL	NULL
dedhara	ahmedabad	5	2022-03-12	2029	karan	50
shikimaru	vadodara	40	2023-04-13	2049	donda	56
shikimaru	vadodara	26	2022-07-19	2079	NULL	NULL
shikimaru	vadodara	6	2023-04-13	2049	donda	56
pain	surat	41	2022-05-16	2509	smith	250
pain	surat	27	2022-08-22	2809	pratik	587
	surat	7	2022-05-16	2509	smith	250

Console

12. Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers

SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", b.city FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id ORDER BY b.salesman_id;

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

cust_name	city	grade	Salesman	city
naruto	delhi	A	raj	vadodra
konahamaru	surat	C	satish	dilhi
dedhara	ahmedabad	B	satish	dilhi
sakura	surat	B	satish	dilhi
shikimaru	vadodara	C	chirag	sura
susake	delhi	C	chirag	sura
suguru	gaziabad	A	akhil	vadodra
itachi	mumbai	A	akhil	vadodra
NULL	NULL	NULL	karan	ahmedabad
gojo	mumbai	A	donda	rajkot
rok	delhi	B	smith	patan
NULL	NULL	NULL	aniket	bhavnagar
NULL	NULL	NULL	manish	jamu
NULL	NULL	NULL	vrushank	manipur
NULL	NULL	NULL	rutvik	anand
NULL	NULL	NULL	sanjay	amreli
NULL	NULL	NULL	kevin	pune
NULL	NULL	NULL	rohan	gandhingar
Console	delhi	A	pratik	raethan

13. write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

Showing rows 1 - 24 (24 total, Query took 0.0000 seconds.)

SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", c.ord_no, c.ord_date, c.purch_amt FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id RIGHT OUTER JOIN orders c ON c.customer_id=a.customer_id;

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

1 | > >> | ☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

cust_name	city	grade	Salesman	ord_no	ord_date	purch_amt
naruto	delhi	A	raj	1	2023-01-12	209
sakura	surat	B	satish	2	2022-10-12	29
susake	delhi	C	chirag	3	2022-11-12	1209
itachi	mumbai	A	akhil	4	2020-02-22	2039
dedhara	ahmedabad	B	satish	5	2022-03-12	2029
shikimaru	vadodara	C	chirag	6	2023-04-13	2049
NULL	NULL	NULL	NULL	7	2022-05-16	2509
goraya	delhi	B	pratik	8	2021-06-17	2609
konahamaru	surat	C	satish	9	2022-07-19	2079
gojo	mumbai	A	donda	10	2022-08-22	2809
saratu	delhi	B	rakesh	11	2022-09-27	2709
NULL	NULL	NULL	NULL	12	2023-10-13	29
suguru	gaziabad	A	akhil	13	2021-12-17	9
NULL	NULL	NULL	NULL	14	2021-11-02	509
NULL	NULL	NULL	NULL	15	2021-07-15	279
deisuke	delhi	A	pratik	16	2020-06-18	249
hunter	gaziabad	B	rakesh	17	2022-05-19	239
Console	delhi	A	pratik	18	2020-04-15	219

14. Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customers. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", c.ord_no, c.ord_date, c.purch_amt FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id LEFT OUTER JOIN orders c ON c.customer_id=a.customer_id WHERE c.purch_amt >= 2000 AND a.grade IS NOT NULL;

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

cust_name	city	grade	Salesman	ord_no	ord_date	purch_amt
itachi	mumbai	A	akhil	4	2020-02-22	2039
dedhara	ahmedabad	B	satish	5	2022-03-12	2029
shikimaru	vadodara	C	chirag	6	2023-04-13	2049
loraya	delhi	B	pratik	8	2021-06-17	2609
konahamaru	surat	C	satish	9	2022-07-19	2079
gojo	mumbai	A	donda	10	2022-08-22	2809
saratu	delhi	B	rakesh	11	2022-09-27	2709
naruto	delhi	A	raj	21	2020-02-22	2039
sakura	surat	B	satish	22	2022-03-12	2029
susake	delhi	C	chirag	23	2023-04-13	2049
itachi	mumbai	A	akhil	24	2022-05-16	2509
dedhara	ahmedabad	B	satish	25	2021-06-17	2609
shikimaru	vadodara	C	chirag	26	2022-07-19	2079
saratu	delhi	B	rakesh	28	2022-09-27	2709
itachi	mumbai	A	akhil	38	2020-02-22	2039
dedhara	ahmedabad	B	satish	39	2022-03-12	2029
shikimaru	vadodara	C	chirag	40	2023-04-13	2049
Console	delhi	B	pratik	42	2021-06-17	2609

15. Write a SQL statement to generate a list of all the salesmen who either work for one or more customers or have yet to join any of them. The customer may have placed one or more orders at or above order amount 2000, and must have a grade, or he may not have placed any orders to the associated supplier

Showing rows 0 - 20 (21 total, Query took 0.0009 seconds)

SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", c.ord_no, c.ord_date, c.purch_amt FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id LEFT OUTER JOIN orders c ON c.customer_id=a.customer_id WHERE c.purch_amt >= 2000 AND a.grade IS NOT NULL;

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

cust_name	city	grade	Salesman	ord_no	ord_date	purch_amt
itachi	mumbai	A	akhil	4	2020-02-22	2039
dedhara	ahmedabad	B	satish	5	2022-03-12	2029
shikimaru	vadodara	C	chirag	6	2023-04-13	2049
loraya	delhi	B	pratik	8	2021-06-17	2609
konahamaru	surat	C	satish	9	2022-07-19	2079
gojo	mumbai	A	donda	10	2022-08-22	2809
saratu	delhi	B	rakesh	11	2022-09-27	2709
naruto	delhi	A	raj	21	2020-02-22	2039
sakura	surat	B	satish	22	2022-03-12	2029
susake	delhi	C	chirag	23	2023-04-13	2049
itachi	mumbai	A	akhil	24	2022-05-16	2509
dedhara	ahmedabad	B	satish	25	2021-06-17	2609
shikimaru	vadodara	C	chirag	26	2022-07-19	2079
saratu	delhi	B	rakesh	28	2022-09-27	2709
itachi	mumbai	A	akhil	38	2020-02-22	2039
dedhara	ahmedabad	B	satish	39	2022-03-12	2029
Console	vadodara	C	chirag	40	2023-04-13	2049

16. Write a SQL statement to generate a report with the customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who neither is on the list nor has a grade.

Server: MySQL-3306 » Database: assinment(sql) » Table: Product

Browse Structure SQL Search Insert Export Import Privileges Operations

Show query box

Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.) [UnitPrice: 10400... - 85...] [ProductName: LAPTOP... - TANDOORI...]

```
SELECT ProductName, UnitPrice FROM Product ORDER BY UnitPrice DESC, ProductName ASC LIMIT 0,10;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

	ProductName	UnitPrice
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	laptop	10400
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	tablet	4999
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	keybord	550
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	sim-card	450
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	watch	450
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	mouse	250
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	beru	156
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	sasuke	156
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	mobile	130
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	tandoori	85

☐ Check all With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

Query results operations

☐ Print Console ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

17. Write a SQL query to combine each row of the salesman table with each row of the customer table

Showing rows 0 - 10 (11 total, 0 in query, query took 0.0000 seconds.)

```
SELECT * FROM salesman a CROSS JOIN customer b;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Salesman_id	Name	City	Commission	customer_id	cust_name	city	grade	salesman_id
144	sagar	amreli	120	1	naruto	delhi	A	123
162	ravi	vadodra	584	1	naruto	delhi	A	123
161	nano	rakol	630	1	naruto	delhi	A	123
153	rakesh	bhalar	95	1	naruto	delhi	A	123
143	pratik	rajsthan	587	1	naruto	delhi	A	123
136	rohan	gandhingar	250	1	naruto	delhi	A	123
135	kevin	pune	310	1	naruto	delhi	A	123
134	sanjay	amreli	120	1	naruto	delhi	A	123
133	rutvik	anand	584	1	naruto	delhi	A	123
132	vrushank	manipur	630	1	naruto	delhi	A	123
131	manish	jamu	95	1	naruto	delhi	A	123
130	aniket	bhavnagar	587	1	naruto	delhi	A	123
129	smith	patan	250	1	naruto	delhi	A	123
128	donda	rajkot	56	1	naruto	delhi	A	123
127	karan	ahmedabad	50	1	naruto	delhi	A	123
126	akhil	vadodra	10	1	naruto	delhi	A	123
125	chirag	sura	100	1	naruto	delhi	A	123
124	satish	dilhi	101	1	naruto	delhi	A	123

Console

18. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for all customers and vice versa for that salesperson who belongs to that city

Showing rows 0 - 18 (19 total, 0 in query, Query took 0.0006 seconds.)

```
SELECT * FROM salesman a CROSS JOIN customer b WHERE a.city IS NOT NULL AND b.grade IS NOT NULL;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Salesman_id	Name	City	Commission	customer_id	cust_name	city	grade	salesman_id
144	sagar	amreli	120	1	naruto	delhi	A	123
162	ravi	vadodra	584	1	naruto	delhi	A	123
161	nano	rakol	630	1	naruto	delhi	A	123
153	rakesh	bhalar	95	1	naruto	delhi	A	123
143	pratik	rajsthan	587	1	naruto	delhi	A	123
136	rohan	gandhingar	250	1	naruto	delhi	A	123
135	kevin	pune	310	1	naruto	delhi	A	123
134	sanjay	amreli	120	1	naruto	delhi	A	123
133	rutvik	anand	584	1	naruto	delhi	A	123
132	vrushank	manipur	630	1	naruto	delhi	A	123
131	manish	jamu	95	1	naruto	delhi	A	123
130	aniket	bhavnagar	587	1	naruto	delhi	A	123
129	smith	patan	250	1	naruto	delhi	A	123
128	donda	rajkot	56	1	naruto	delhi	A	123
127	karan	ahmedabad	50	1	naruto	delhi	A	123
126	akhil	vadodra	10	1	naruto	delhi	A	123
125	chirag	sura	100	1	naruto	delhi	A	123

Console

19. Write a SQL statement to create a Cartesian product between salesperson and customer, i.e. each salesperson will appear for every customer and vice versa for those salesmen who belong to a city and customers who require a grade

```
SELECT * FROM salesman a CROSS JOIN customer b WHERE a.city IS NOT NULL AND b.grade IS NOT NULL;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Salesman_id	Name	City	Commission	customer_id	cust_name	city	grade	salesman_id
144	sagar	amreli	120	1	naruto	delhi	A	123
162	ravi	vadodra	584	1	naruto	delhi	A	123
161	nano	rakol	630	1	naruto	delhi	A	123
153	rakesh	bhalar	95	1	naruto	delhi	A	123
143	pratik	rajsthan	587	1	naruto	delhi	A	123
136	rohan	gandhingar	250	1	naruto	delhi	A	123
135	kevin	pune	310	1	naruto	delhi	A	123
134	sanjay	amreli	120	1	naruto	delhi	A	123
133	rutvik	anand	584	1	naruto	delhi	A	123
132	vrushank	manipur	630	1	naruto	delhi	A	123
131	manish	jamu	95	1	naruto	delhi	A	123
130	aniket	bhavnagar	587	1	naruto	delhi	A	123
129	smith	patan	250	1	naruto	delhi	A	123
128	donda	rajkot	56	1	naruto	delhi	A	123
127	karan	ahmedabad	50	1	naruto	delhi	A	123
126	akhil	vadodra	10	1	naruto	delhi	A	123
125	chirag	sura	100	1	naruto	delhi	A	123
124	satish	dilhi	101	1	naruto	delhi	A	123

Console

20. Write a SQL statement to make a Cartesian product between salesman and customer i.e. each salesman will appear for all customers and vice versa for those salesmen who must belong to a city which is not the same as his customer and the customers should have their own grade

Browse	Structure	SQL	Search	Insert	Export	Import	Privileges	Operations	Triggers
SELECT * FROM salesman a CROSS JOIN customer b WHERE a.city IS NOT NULL AND b.grade IS NOT NULL AND a.city<>b.city;									
<input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]									
<input type="checkbox"/> Show all Number of rows: 25 Filter rows: Search this table									
Extra options									
Salesman_id	Name	City	Commission	customer_id	cust_name	city	grade	salesman_id	
144	sagar	amreli	120	1	naruto	delhi	A	123	
162	ravi	vadodra	584	1	naruto	delhi	A	123	
161	nano	rakol	630	1	naruto	delhi	A	123	
153	rakesh	bhalar	95	1	naruto	delhi	A	123	
143	pratik	rajsthan	587	1	naruto	delhi	A	123	
136	rohan	gandhingar	250	1	naruto	delhi	A	123	
135	kevin	pune	310	1	naruto	delhi	A	123	
134	sanjay	amreli	120	1	naruto	delhi	A	123	
133	rutvik	anand	584	1	naruto	delhi	A	123	
132	vrushank	manipur	630	1	naruto	delhi	A	123	
131	manish	jamu	95	1	naruto	delhi	A	123	
130	aniket	bhavnagar	587	1	naruto	delhi	A	123	
129	smith	patan	250	1	naruto	delhi	A	123	
128	donda	rajkot	56	1	naruto	delhi	A	123	
127	karan	ahmedabad	50	1	naruto	delhi	A	123	
126	akhil	vadodra	10	1	naruto	delhi	A	123	
125	chirag	sura	100	1	naruto	delhi	A	123	
124	satish	dilhi	101	1	naruto	delhi	A	123	
123	raj	vadodra	100	1	naruto	delhi	A	123	