Java Enterprise Edition

ANUDIP FOUNDATION

SQL JOINS: Exercise-1 Time: 240 Min

Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.

Sample table: salesman

```
 salesman_id |    name    |   city    | commission
-------------+------------+-----------+------------
       5001 | James Hoog | New York |       0.15
       5002 | Nail Knite | Paris    |       0.13
       5005 | Pit Alex   | London   |       0.11
       5006 | Mc Lyon    | Paris    |       0.14
       5007 | Paul Adam  | Rome     |       0.13
       5003 | Lauson Hen | San Jose |       0.12
```

Sample table: customer

```
 customer_id |   cust_name    |    city    | grade | salesman_id
-------------+----------------+------------+-------+-------------
       3002 | Nick Rimando   | New York   |   100 |       5001
       3007 | Brad Davis     | New York   |   200 |       5001
       3005 | Graham Zusi    | California |   200 |       5002
       3008 | Julian Green   | London     |   300 |       5002
       3004 | Fabian Johnson | Paris      |   300 |       5006
       3009 | Geoff Cameron  | Berlin     |   100 |       5003
       3003 | Jozy Altidor   | Moscow     |   200 |       5007
       3001 | Brad Guzan     | London     |       |       5005
```

Sample Solution:

```
SELECT salesman.name AS "Salesman",

customer.cust_name, customer.city

FROM salesman,customer

WHERE salesman.city=customer.city;
```

Output of the Query:

```
Salesman        cust_name               city
James Hoog      Nick Rimando             New York
James Hoog      Brad Davis               New York
Pit Alex  Julian Green              London
```

```
Mc Lyon  Fabian Johnson          Paris
Nail Knite           Fabian Johnson        Paris
Pit Alex   Brad Guzan                  London
```

2. Write a SQL statement to make a list with order no, purchase amount, customer name and their cities for those orders which order amount between 500 and 2000. Go to the editor

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005

Sample Solution:

```
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;
```

Output of the Query:

```
ord_no    purch_amt        cust_name              city
70007     948.50  Graham Zusi       California
70010    1983.43 Fabian Johnson     Paris
```

SQL JOINS: Exercise-3

Write a SQL statement to know which salesman are working for which customer.

Sample table: customer

```
 customer_id |   cust_name    |    city    | grade | salesman_id
-------------+----------------+------------+-------+-------------
       3002 | Nick Rimando   | New York   |   100 |       5001
       3007 | Brad Davis     | New York   |   200 |       5001
       3005 | Graham Zusi    | California |   200 |       5002
       3008 | Julian Green   | London     |   300 |       5002
       3004 | Fabian Johnson | Paris      |   300 |       5006
       3009 | Geoff Cameron  | Berlin     |   100 |       5003
       3003 | Jozy Altidor   | Moscow     |   200 |       5007
       3001 | Brad Guzan     | London     |       |       5005
```

Sample table: salesman

```
salesman_id |    name    |   city   | commission
-------------+------------+----------+------------
       5001 | James Hoog | New York |       0.15
       5002 | Nail Knite | Paris    |       0.13
       5005 | Pit Alex   | London   |       0.11
       5006 | Mc Lyon    | Paris    |       0.14
       5007 | Paul Adam  | Rome     |       0.13
       5003 | Lauson Hen | San Jose |       0.12
```

Sample Solution:

```
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;
```

Output of the Query:

Customer Name	city	Salesman	commission
Nick Rimando	New York	James Hoog	0.15
Brad Davis	New York	James Hoog	0.15
Graham Zusi	California	Nail Knite	0.13
Julian Green	London	Nail Knite	0.13
Fabian Johnson	Paris	Mc Lyon	0.14
Geoff Cameron	Berlin	Lauson Hen	0.12
Jozy Altidor	Moscow	Paul Adam	0.13
Brad Guzan	London	Pit Alex	0.11

SQL JOINS: Exercise-4

Write a SQL statement to find the list of customers who appointed a salesman for their jobs who gets a commission from the company is more than 12%.

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		

Sample table: salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

Sample Solution:

```
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;
```

Output of the Query:

```
Customer Name   city              Salesman          commission
Nick Rimando    New York              James Hoog    0.15
Brad Davis      New York              James Hoog    0.15
Graham Zusi     California            Nail Knite    0.13
Julian Green    London              Nail Knite      0.13
Fabian Johnson Paris            Mc Lyon           0.14
Jozy Altidor        Moscow              Paul Adam       0.13
```

SQL JOINS: Exercise-5

Write a SQL statement to find the list of customers who appointed a salesman for their jobs who does not live in the same city where their customer lives, and gets a commission is above 12%.

Sample table: customer

```
customer_id |   cust_name    |    city    | grade | salesman_id
------------+----------------+------------+-------+-------------
       3002 | Nick Rimando   | New York   |   100 |       5001
       3007 | Brad Davis     | New York   |   200 |       5001
       3005 | Graham Zusi    | California |   200 |       5002
       3008 | Julian Green   | London     |   300 |       5002
       3004 | Fabian Johnson | Paris      |   300 |       5006
       3009 | Geoff Cameron  | Berlin     |   100 |       5003
       3003 | Jozy Altidor   | Moscow     |   200 |       5007
       3001 | Brad Guzan     | London     |       |       5005
```

Sample table: salesman

```
salesman_id |    name    |   city   | commission
------------+------------+----------+------------
       5001 | James Hoog | New York |       0.15
       5002 | Nail Knite | Paris    |       0.13
       5005 | Pit Alex   | London   |       0.11
       5006 | Mc Lyon    | Paris    |       0.14
       5007 | Paul Adam  | Rome     |       0.13
```

5003 | Lauson Hen | San Jose | 0.12

Sample Solution:

```
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;
```

Output of the Query:

```
Customer Name   city          Salesman      city    commission
Graham Zusi     California    Nail Knite    Paris   0.13
Julian Green    London              Nail Knite      Paris   0.13
Jozy Altidor    Moscow        Paul Adam     Rome    0.13
```

SQL JOINS: Exercise-6

Write a SQL statement to find the details of a order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and commission rate he gets for an order.

Sample table: orders

```
ord_no     purch_amt   ord_date    customer_id  salesman_id
---------- ---------- ---------- ----------- -----------
70001      150.5       2012-10-05  3005        5002
70009      270.65      2012-09-10  3001        5005
70002      65.26       2012-10-05  3002        5001
70004      110.5       2012-08-17  3009        5003
70007      948.5       2012-09-10  3005        5002
70005      2400.6      2012-07-27  3007        5001
70008      5760        2012-09-10  3002        5001
70010      1983.43     2012-10-10  3004        5006
70003      2480.4      2012-10-10  3009        5003
70012      250.45      2012-06-27  3008        5002
70011      75.29       2012-08-17  3003        5007
70013      3045.6      2012-04-25  3002        5001
```

Sample table: customer

```
customer_id |   cust_name    |    city    | grade | salesman_id
------------+----------------+------------+-------+-------------
       3002 | Nick Rimando   | New York   |   100 |       5001
       3007 | Brad Davis     | New York   |   200 |       5001
       3005 | Graham Zusi    | California |   200 |       5002
       3008 | Julian Green   | London     |   300 |       5002
       3004 | Fabian Johnson | Paris      |   300 |       5006
       3009 | Geoff Cameron  | Berlin     |   100 |       5003
       3003 | Jozy Altidor   | Moscow     |   200 |       5007
       3001 | Brad Guzan     | London     |       |       5005
```

Sample table: salesman

```
salesman_id |    name    |   city    | commission
------------+------------+-----------+------------
       5001 | James Hoog | New York  |       0.15
       5002 | Nail Knite | Paris     |       0.13
       5005 | Pit Alex   | London    |       0.11
       5006 | Mc Lyon    | Paris     |       0.14
       5007 | Paul Adam  | Rome      |       0.13
       5003 | Lauson Hen | San Jose  |       0.12
```

Sample Solution:

```
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;
```

Output of the Query:

ord_no	ord_date	purch_amt	Customer Name	grade	Salesman	commission
70009	2012-09-10	270.65	Brad Guzan		Pit Alex	0.11
70002	2012-10-05	65.26	Nick Rimando	100	James Hoog	0.15
70004	2012-08-17	110.50	Geoff Cameron	100	Lauson Hen	0.12
70005	2012-07-27	2400.60	Brad Davis	200	James Hoog	0.15
70008	2012-09-10	5760.00	Nick Rimando	100	James Hoog	0.15
70010	2012-10-10	1983.43	Fabian Johnson	300	Mc Lyon	0.14
70003	2012-10-10	2480.40	Geoff Cameron	100	Lauson Hen	0.12
70011	2012-08-17	75.29	Jozy Altidor	200	Paul Adam	0.13
70013	2012-04-25	3045.60	Nick Rimando	100	James Hoog	0.15
70001	2012-10-05	150.50	Graham Zusi	200	Nail Knite	0.13
70007	2012-09-10	948.50	Graham Zusi	200	Nail Knite	0.13
70012	2012-06-27	250.45	Julian Green	300	Nail Knite	0.13

SQL JOINS: Exercise-7

Write a SQL statement to make a join on the tables salesman, customer and orders in such a form that the same column of each table will appear once and only the relational rows will come.

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005

Sample table : salesman

```
salesman_id |    name    |   city    | commission
-------------+------------+----------+------------
       5001 | James Hoog | New York |       0.15
       5002 | Nail Knite | Paris    |       0.13
       5005 | Pit Alex   | London   |       0.11
       5006 | Mc Lyon    | Paris    |       0.14
       5007 | Paul Adam  | Rome     |       0.13
       5003 | Lauson Hen | San Jose |       0.12
```

Sample Solution:

```
SELECT *

FROM orders

NATURAL JOIN customer

NATURAL JOIN salesman;
```

Output of the Query:

```
salesman_id city customer_id       ord_no purch_amt ord_date  cust_name       grade   name
commission
5005     London   3001     70009  270.65    2012-09-10 Brad Guzan      Pit Alex          0.11
5001     New York         3002     70002   65.26    2012-10-05 Nick Rimando 100 James Hoog  0.15
5001     New York         3007     70005  2400.60   2012-07-27 Brad Davis   200 James Hoog
         0.15
5001     New York         3002     70008  5760.00   2012-09-10 Nick Rimando 100 James Hoog
         0.15
5006     Paris     3004     70010  1983.43   2012-10-10 Fabian Johnson300 Mc Lyon      0.14
5001     New York         3002     70013  3045.60   2012-04-25 Nick Rimando  100 James Hoog
         0.15
```

SQL JOINS: Exercise-8

Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005

Sample table: salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

Sample Solution:

```
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;
```

Output of the Query:

cust_name	city	grade	Salesman	city
Brad Guzan	London		Pit Alex	London
Nick Rimando	New York	100	James Hoog	New York
Jozy Altidor	Moscow	200	Paul Adam	Rome
Fabian Johnson	Paris	300	Mc Lyon	Paris
Graham Zusi	California	200	Nail Knite	Paris
Brad Davis	New York	200	James Hoog	New York
Julian Green	London	300	Nail Knite	Paris
Geoff Cameron	Berlin	100	Lauson Hen	San Jose

SQL JOINS: Exercise-9

Write a SQL statement to make a list in ascending order for the customer who holds a grade less than 300 and works either through a salesman or by own.

Sample table: customer

```
customer_id |   cust_name    |    city    | grade | salesman_id
------------+----------------+------------+-------+-------------
       3002 | Nick Rimando   | New York   |   100 |       5001
       3007 | Brad Davis     | New York   |   200 |       5001
       3005 | Graham Zusi    | California |   200 |       5002
       3008 | Julian Green   | London     |   300 |       5002
       3004 | Fabian Johnson | Paris      |   300 |       5006
       3009 | Geoff Cameron  | Berlin     |   100 |       5003
       3003 | Jozy Altidor   | Moscow     |   200 |       5007
       3001 | Brad Guzan     | London     |       |       5005
```

Sample table: salesman

```
salesman_id |    name    |   city   | commission
------------+------------+----------+------------
       5001 | James Hoog | New York |       0.15
       5002 | Nail Knite | Paris    |       0.13
       5005 | Pit Alex   | London   |       0.11
       5006 | Mc Lyon    | Paris    |       0.14
       5007 | Paul Adam  | Rome     |       0.13
       5003 | Lauson Hen | San Jose |       0.12
```

Sample Solution:

```
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;
```

Output of the Query:

cust_name	city	grade	Salesman	city
Nick Rimando	New York	100	James Hoog	New York
Jozy Altidor	Moscow	200	Paul Adam	Rome
Graham Zusi	California	200	Nail Knite	Paris

Brad Davis	New York	200	James Hoog	New York
Geoff Cameron	Berlin	100	Lauson Hen	San Jose

SQL JOINS: Exercise-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 find that either any of the existing customers have placed no order or placed one or more orders.

Sample table: orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001
70008	5760	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70003	2480.4	2012-10-10	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007
70013	3045.6	2012-04-25	3002	5001

Sample table: customer

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3007	Brad Davis	New York	200	5001
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3004	Fabian Johnson	Paris	300	5006
3009	Geoff Cameron	Berlin	100	5003
3003	Jozy Altidor	Moscow	200	5007
3001	Brad Guzan	London		5005

Sample Solution:

```
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;
```

Output of the Query:

cust_name	city	ord_no	ord_date	Order Amount
Nick Rimando	New York	70013	2012-04-25	3045.60
Julian Green	London	70012	2012-06-27	250.45
Brad Davis	New York	70005	2012-07-27	2400.60
Jozy Altidor	Moscow	70011	2012-08-17	75.29
Geoff Cameron	Berlin	70004	2012-08-17	110.50
Brad Guzan	London	70009	2012-09-10	270.65
Nick Rimando	New York	70008	2012-09-10	5760.00
Graham Zusi	California	70007	2012-09-10	948.50
Graham Zusi	California	70001	2012-10-05	150.50
Nick Rimando	New York	70002	2012-10-05	65.26
Fabian Johnson	Paris	70010	2012-10-10	1983.43
Geoff Cameron	Berlin	70003	2012-10-10	2480.40