**01 Feb 2016:**

* create table custo(

custo\_id number(10),

custo\_name varchar2(20) not null,

age number(4),

course varchar2(15),

register\_date date);

* insert into custo(custo\_id, custo\_name, age, course, register\_date)

values(&custo\_id, '&custo\_name', &age, '&course', '&register\_date');

* Alter table custo

add email varchar2(25) unique;

* alter table custo

modify register\_date date default sysdate;

* alter table custo

modify age number(4) check(age>=18);

* alter table custo

modify age number(9);

* create sequence custo\_id\_seq

start with 100 increment by 10 maxvalue 10000;

* create sequence custo\_id\_sequ

start with 100 increment by 10 maxvalue 10000

nocache

nocycle;

* alter table custo

add constraint custo\_id\_pk primary key (CUSTO\_ID);

* create table customer\_order(

order\_id number(10),

order\_date date,

product\_name varchar2(15),

price number(10,12),

quantity number(4),

CUSTO\_ID number(10),

constraint order\_id\_pk primary key(order\_id),

constraint custo\_id\_fk foreign key(custo\_id) references custo(CUSTO\_ID)

);

* insert into customer\_order

values(1, sysdate, 'pant', 550, 10, 101);

insert into customer\_order

values(2, sysdate, 'pant', 450, 10, 102);

insert into customer\_order

values(3, sysdate, 'pant', 350, 10, 103);

insert into customer\_order

values(4, sysdate, 'pant', 250, 10, 104);

insert into customer\_order

values(5, sysdate, 'pant', 150, 10, 105);

insert into customer\_order

values(6, sysdate, 'shirt', 500, 10, 101);

insert into customer\_order

values(7, sysdate, 'shirt', 400, 10, 102);

insert into customer\_order

values(8, sysdate, 'shirt', 300, 10, 103);

insert into customer\_order

values(9, sysdate, 'shirt', 200, 10, 104);

insert into customer\_order

values(10, sysdate, 'shirt', 100, 10, 105);

insert into customer\_order

values(11, sysdate, 'shoes', 1000, 10, 101);

insert into customer\_order

values(12, sysdate, 'shoes', 900, 10, 102);

insert into customer\_order

values(13, sysdate, 'shoes', 800, 10, 103);

insert into customer\_order

values(14, sysdate, 'shoes', 700, 10, 104);

insert into customer\_order

values(15, sysdate, 'shoes', 600, 10, 105);

* update customer\_order

set status='paid'

where order\_id in(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

* select custo\_id, QUANTITY\*PRICE total, custo\_name

from customer\_order natural join custo;

* select sum(QUANTITY\*PRICE) total, custo\_name

from customer\_order natural join custo

group by CUSTO\_NAME;

* select QUANTITY\*PRICE, custo\_name, product\_name

from customer\_order natural join custo

where product\_name not in(select product\_name from customer\_order natural join custo where custo\_name='reza')

* select QUANTITY\*PRICE, custo\_name, product\_name

from customer\_order natural join custo

where product\_name in(select product\_name from customer\_order natural join custo where custo\_name='reza')

and custo\_name <>'reza'

* select order\_id, product\_name, quantity, price

from customer\_order e

where quantity>(select avg(quantity) from customer\_order where product\_name=e.product\_name)

* select product\_name, custo\_id, sum(quantity)

from customer\_order

group by rollup (custo\_id, product\_name);

**08 Feb 2016 solution:**

1. create table customer\_sales(

order\_no number

constraint cust\_sal\_ord\_no\_pk primary key,

order\_name varchar2(25) not null,

product\_name varchar2(20) not null,

price number(12,2),

quantity number(10),

order\_date date

);

1. create sequence ord\_no\_seq start with 1000 increment by 10;
2. alter table customer\_sales add tax number(3,2);
3. alter table customer\_sales modify order\_date date default sysdate;
4. alter table customer\_sales modify product\_name varchar2(25);
5. insert into customer\_sales (order\_no, order\_name, product\_name, price, quantity, order\_date, tax)

values(ORD\_NO\_SEQ.nextval,'&order\_name', '&product\_name', &price, &quantity, &order\_date, &tax);

**Query:**

* select order\_no, order\_name, product\_name, quantity, price, tax, price\*quantity+tax\*100 total

from customer\_sales;

* create view customer\_sales\_view As select order\_no, order\_name, product\_name, quantity, price, tax, price\*quantity+tax\*100 total

from customer\_sales;

* select order\_no, max(quantity), min(quantity)

from customer\_sales

group by order\_no;

* select order\_no, product\_name, sum(quantity)

from customer\_sales

group by rollup(order\_no, product\_name

* select order\_no, product\_name, quantity, price

from customer\_sales

where quantity>(select avg(quantity) from customer\_sales);

* update customer\_sales

set quantity=(select quantity from customer\_sales where order\_no=1030)

where order\_no=1020;