* select EMPLOYEE\_ID,LAST\_NAME, SALARY\*.4 medical, SALARY, SALARY\*.6 houserent,

commission\_pct, salary+SALARY\*.6+SALARY\*.4+salary\*commission\_pct total

from employees;

* select last\_name,salary,12\*salary+100

from employees;

* select last\_name,salary,12\*(salary+100)

from employees;

* select last\_name ||' is a '||job\_id As "Employee Details"

from employees;

* select department\_name|| q'[, it's assigned Manager Id: ]' || manager\_id

As "Department and Manager"

from departments;

* select distinct department\_id, job\_id

from employees;

* select \*from employees

where department\_id=90;

* select \*from employees

where department\_id=90 and (salary=17000 or department\_id=80);

* select last\_name, job\_id, department\_id

from employees

where last\_name='Whalen';

* select last\_name, salary

from employees

where salary<=3000;

* select employee\_id, last\_name, salary, manager\_id

from employees

where manager\_id in (100,101,201);

* select last\_name, job\_id, department\_id,hire\_date, salary\*12 annsal

from employees

order by annsal ;

* select employee\_id, last\_name, job\_id, && column\_name

from employees

order by & column\_name;

* select upper(last\_name), lower(first\_name)

from employees;

* select first\_name, last\_name, concat(first\_name, last\_name),

substr(last\_name,2,2), instr(first\_name,'a')

from employees;

* select employee\_id, concat(first\_name, last\_name) NAME,

length(last\_name), instr(first\_name,'a') "contains'a'?"

from employees;

where substr(last\_name,-1,1)='n';

* select last\_name, LPAD(salary,30,'\*'), RPAD(salary,30,'\*')

from employees;

* select last\_name, trim('n' from last\_name), replace(last\_name, 'a', 'D')

from employees;

* select initcap('sql course') from dual;
* select round(45.926,2), trunc(45.926,2), round(45.926,-1), trunc(45.926,-1)

from dual;

* select employee\_id, mod((sysdate-HIRE\_DATE),365) year, mod((sysdate

HIRE\_DATE),365),30 month, mod(((sysdate-HIRE\_DATE),365),30),24 day

from employees;

* select employee\_id, (sysdate-HIRE\_DATE)/365 year, mod((sysdate-HIRE\_DATE),365) day

from employees;

* select employee\_id, (sysdate-HIRE\_DATE)/365 year, mod((sysdate-HIRE\_DATE),365)/30 month, mod(mod((sysdate-HIRE\_DATE),365),30) day

from employees;

* select employee\_id, hire\_date, months\_between(sysdate,hire\_date) tenure, add\_months(hire\_date,6) review, next\_day (hire\_date, 'FRIDAY'), last\_day (hire\_date)

from employees;

* select employee\_id, hire\_date, round(hire\_date, 'MONTH'), round(hire\_date, 'YEAR'), trunc(hire\_date,'MONTH')

from employees;

* select employee\_id, to\_char(hire\_date,'dy/MM/year') month\_hired

from employees;

* select employee\_id, to\_char(hire\_date,'ddspts/MM/year') month\_hired

from employees;

* select last\_name, to\_char(hire\_date, 'fmDD Month YYYY')

AS HIREDATE

from employees;

* select last\_name, to\_char(hire\_date, 'fmDdspth "of" Month YYYY fmHH:MI:SS PM')

HIREDATE

from employees;

* select employee\_id, salary, to\_char(salary,'0,00,000.00'), to\_char(salary,'9,99,999.99')

from employees;

* select employee\_id, salary, to\_char(salary,'$0,00,000.00'), to\_char(salary,'$9,99,999.99')

from employees;

* select employee\_id, salary,last\_name, hire\_date

from employees

where hire\_date=to\_date('may 24,1999','fmMonth DD, YYYY');

* select last\_name, salary, NVL(commission\_pct,0), (salary\*12)+(salary\*12\*NVL(commission\_pct,0)) AN\_SAL

from employees;

* select last\_name, salary, commission\_pct,

NVL2 (commission\_pct, 'SAL+COMM', 'SAL') income

from employees

Where department\_id IN (50,80);

* select last\_name, salary, commission\_pct, NVL2(commission\_pct, salary+salary\*commission\_pct, salary)

from employees;

* select first\_name, length(first\_name) "expr1",

last\_name,length(last\_name) "expr2", nullif(length(first\_name),length(last\_name)) result

from employees;

* select last\_name, manager\_id, commission\_pct,coalesce(manager\_id,commission\_pct,-1) comm

from employees

order by commission\_pct;

select last\_name, job\_id, salary,

CASE job\_id WHEN 'it\_prog' THEN 1.10\*salary

WHEN 'st\_cleark' THEN 1.15\*salary

WHEN 'sa\_rep' THEN 1.20\*salary

ELSE salary END "REVISED\_SALARY"

FROM employees;

* select last\_name, job\_id, salary,

(CASE when salary<5000 THEN 'low'

when salary<10000 THEN 'medium'

when salary<20000 THEN 'high'

else 'excellent'

END) qualified\_salary

FROM employees;

* select

COUNT(\*)

FROM employees

WHERE DEPARTMENT\_ID = 50;

* SELECT COUNT(\*), avg(salary), max(salary), min(salary), sum(salary)

FROM employees

WHERE JOB\_ID LIKE '%REP%'

* SELECT COUNT(COMMISSION\_PCT)

FROM employees

WHERE DEPARTMENT\_ID=80;

* SELECT COUNT(DISTINCT department\_id)

FROM employees;

* SELECT AVG(NVL(commission\_pct,0))

FROM employees;

* SELECT department\_id dept\_id, job\_id, sum(salary)

from employees

Group by department\_id, job\_id;

* SELECT department\_id dept\_id, job\_id, sum(salary)

from employees

where salary<10000

Group by department\_id, job\_id

order by department\_id, job\_id;

* select last\_name, salary, department\_name

from employees e, departments;

* select last\_name, salary, department\_name

from employees e, departments d

where e.department\_id=d.department\_id;

(left outer joint)

* select last\_name, salary, department\_name

from employees e, departments d

where e.department\_id(+)=d.department\_id;

* select last\_name, salary, department\_name

from employees e left outer join departments d

on (e.department\_id=d.department\_id);

(right outer joint)

* select last\_name, salary, department\_name

from employees e, departments d

where e.department\_id=d.department\_id(+);

* select department\_id, department\_name, location\_id, city

from departments NATURAL JOIN locations;

* select department\_id, department\_name, location\_id, city

from departments NATURAL JOIN locations

where department\_id IN (20,50);

* select last\_name, salary, department\_name

from employees NATURAL JOIN departments;

* select e.employee\_id, e.last\_name, d.location\_id, department\_id

from employees e JOIN departments d

USING (department\_id);

* select e.employee\_id, e.last\_name, e.department\_id, d.department\_id, d.location\_id

from employees e JOIN departments d

on (e.department\_id=d.department\_id);

* select e.last\_name emp, m.last\_name mgr

from employees e JOIN employees m

on (e.manager\_id=m.employee\_id);

* select e.manager\_id, e.employee\_id, e.last\_name, e.department\_id, d.department\_id, d.location\_id

from employees e JOIN departments d

on (e.department\_id=d.department\_id)

and e.manager\_id=149;

* select employee\_id, last\_name, salary, job\_title

from employees e join jobs j

on e.salary between j.min\_salary and j.max\_salary;

* select employee\_id, last\_name, salary, job\_title

from employees e join jobs j

on e.salary between j.min\_salary and j.max\_salary

order by employee\_id;

* select last\_name,salary

from employees

where last\_name='Abel';

* select last\_name, salary

from employees

where salary=11000;

(sub-query)

* select last\_name,salary

from employees

where salary=(select salary from employees where last\_name='Abel');

* select last\_name,salary

from employees

where salary>(select salary from employees where last\_name='Abel');

* select last\_name,salary

from employees

where salary>(select avg(salary) from employees);

* select department\_id, last\_name, salary

from employees

where salary in (select min(salary) from employees group by department\_id);

* select department\_id, last\_name, salary

from employees

where salary in (select min(salary) from employees where department\_id=80);

* select last\_name, job\_id

from employees

where job\_id=(select job\_id from employees where employee\_id=141);

* select last\_name, job\_id, salary

from employees

where job\_id=(select job\_id from employees where employee\_id=141)

and salary>(select salary from employees where employee\_id=143);

* select department\_id, min(salary)

from employees

group by department\_id

having min(salary)>(select min(salary) from employees where department\_id=50);

* select job\_id, avg(salary)

from employees

group by job\_id

having avg(salary)=(select min(avg(salary)) from employees group by job\_id);

* select employee\_id, last\_name, job\_id,salary

from employees

where salary<any(select salary from employees where job\_id='IT\_PROG')

and job\_id<>'IT\_PROG';

* select emp.last\_name

from employees emp

where emp.employee\_id IN

(select mgr.manager\_id

from employees mgr);

* select last\_name, manager\_id

from employees

where employee\_id NOT IN

(select manager\_id

from employees

where manager\_id IS NOT NULL);

* create table result(

exam\_no number(10) primary key, roll\_no number(10), name varchar2(15), subject varchar2(10), mark number(3), exam\_date date

);

* insert into result(exam\_no, roll\_no, name, subject, mark, exam\_date)

values(1,123,'REZA','Database',65,'31-Dec-15')

* insert into result(exam\_no, roll\_no, mark)

values(7,152,87);

* insert into result

values(14,11,null,null,66,sysdate);

* commit;
* update result set name='Fakhrul'
* rollback
* update result set name='Fakhrul'

where exam\_no=14;

* delete from result

where exam\_no=7;

* update employees

set salary=(select salary

from employees

where employee\_id=205),

last\_name=(select last\_name

from employees

where employee\_id=205)

where employee\_id=114;

* update employee27

set salary=(select salary

from employees

where employee\_id=205),

job\_id=(select job\_id

from employees

where employee\_id=205)

where employee\_id=114;

* insert into

(select employee\_id, last\_name, email, hire\_date, job\_id, salary, department\_id

from employee27 where department\_id=50)

values(99999, 'taylor', 'DTAYLOR', to\_date('07-JUN-99', 'DD-MON-RR'), 'ST\_CLEAR', 5000, 50);

* delete from employee27;

savepoint A;

insert into employee27(employee\_id,last\_name,job\_id,salary,email,hire\_date)

values(101,'Rana','AC\_ACCOUNT',5000,'abc',sysdate);

savepoint B;

insert into employee27(employee\_id,last\_name,job\_id,salary,email,hire\_date)

values(102,'Jashim','DC\_ACCOUNT',10000,'dsc',sysdate);

rollback to B;

rollback;

**Book 2**

* create user Zihad identified by R27;
* grant create session to Zihad;
* grant resource, connect to Zihad;
* create table student

(student\_id number(10) primary key,

student\_name varchar2(20) not null,

age number(3),

dob date,

course varchar2(10)

);

* select \* from user\_objects;
* create table time\_example

(order\_date timestamp with local time zone);

insert into time\_example

values(sysdate);

* create table time\_example1

(order\_date timestamp with time zone);

insert into time\_example1

values(sysdate);

* insert into time\_example2

values(interval '320' month(3) );

select \*

from time\_example2;

* create table time\_example3

(day\_duration interval day (3) to second);

insert into time\_example3 (day\_duration)

values (interval '180' day (3));

select sysdate+day\_duration "Half Year"

from time\_example3;

* create table exam

(

exam\_no number(10),

month varchar2(4) not null,

student\_id number(10),

exam\_type varchar2(4) check(exam\_type in('mcq','des''evi')),

mark number(3),

constraint exam\_no\_pk primary key(exam\_no),

constraint exam\_student\_id\_fk foreign key(student\_id) references student(student\_id)

);

* select \* from user\_constraints;
* insert into student

values(101,'Riad',25,'31-aug-1990','J2ee');

insert into student

values(102,'Obaidul',24,'11-jul-1991','c++');

insert into student

values(104,'Imran',28,'11-jun-1990','c#');

insert into student

values(105,'Reza',24,'11-jan-1992','c#');

* insert into exam

values(5, 'feb', 105,'mcq', 48

* alter table exam

drop constraint EXAM\_STUDENT\_ID\_FK ;

* alter table exam

add constraint EXAM\_STUDENT\_ID\_FK

foreign key(student\_id)

references student (student\_id)

on delete cascade ;

* alter table student

add email varchar2(20);

* alter table student

drop column email ;

* create view empvu81

as select employee\_id, last\_name, salary

from employees

where department\_id=80;

* select \* from empvu81;
* create or replace view empvu81

as select employee\_id, last\_name, salary,department\_id

from employees

where department\_id=80;

* create view salvu51

as select employee\_id id\_number, last\_name name, salary\*12 ann\_salary

from employees

where department\_id=50;

* select \* from salvu51;
* create or replace view salvu51

as select employee\_id id\_number, last\_name name, salary\*12 ann\_salary, department\_id

from employees

where department\_id=50;

* create or replace view empvu81

(id\_number, name,sal,department\_id)

as select employee\_id, first\_name||' '||last\_name, salary, department\_id

from employees

where department\_id=80;

* create or replace view dept\_sum\_vu

(name, minsal, maxsal, avgsal)

as select d.department\_name, min(e.salary),max(e.salary),avg(e.salary)

from employees e join departments d

on (e.department\_id=d.department\_id)

group by d.department\_name;

* create or replace view empvu21

as select \*

from employees

where department\_id=20

with check option constraint empvu21\_ck;

* update empvu21

set department\_id=10

where employee\_id=201;

**EXAMPLE-1:**

create table member(

member\_id number(10),

last\_name varchar2(25) not null,

first\_name varchar2(25),

address varchar2(100),

city varchar2(30),

phone varchar2(15),

join\_date date defult sysdate not null,

constraint member\_id\_pk primary key(member\_id )

);

**EXAMPLE-2:**

create table title(

title\_id number(10),

title varchar2(60) not null,

description varchar2(400) not null,

rating varchar2(4) check (rating in('G','PG','R','NC17','NR')),

category varchar2(20) check (rating in('DRAMA','COMEDY','ACTION','CHILD','SCIFI','DOCUMENT TARY')),

release\_date (date),

constraint title\_id\_pk primary key(title\_id )

);

**EXAMPLE-3:**

create table title\_copy(

copy\_id number(10),

title\_id number(10),

status varchar2(15) not null,

constraint copy\_id\_title\_id\_pk primary key(copy\_id, title\_id),

constraint status\_check\_ck check(status in('AVAILABLE','DESTROYED','RENTED','RESERVED')),

constraint title\_id\_fk foreign key(title\_id) references title(title\_id)

);

**EXAMPLE-4:**

create table rental(

book\_date date default sysdate,

member\_id number(10),

copy\_id number(10),

act\_ret\_date date,

exp\_ret\_date date default sysdate+2,

title\_id number(10),

constraint book\_date\_pk primary key(book\_date,member\_id,copy\_id,title\_id),

constraint member\_id\_fk1 foreign key(member\_id) references member(member\_id ),

constraint copy\_id\_title\_id\_fk2 foreign key(copy\_id,title\_id) references title\_copy(copy\_id,title\_id)

);

**EXAMPLE-5:**

create table reservation(

res\_date date,

member\_id number(10),

title\_id number(10),

constraint reservation\_res\_date\_pk primary key(res\_date,member\_id,title\_id),

constraint reservation\_member\_id\_fk1 foreign key(member\_id) references member(member\_id),

constraint reservation\_title\_id\_fk2 foreign key(title\_id) references title(title\_id)

);

**SEQUENCE-3(A):**

* create sequence member\_id\_seq

start with 101 nocache

increment by 1;

* select \* from user\_sequences;

**SEQUENCE-3(B):**

* create sequence title\_id\_seq

start with 92 nocache

increment by 1;

**DATA INPUT-4(a):**

* insert into member

values(member\_id\_seq.nextval,'Carmen','Velasquez','283 King Street','Seattle','206-899-6666','08-MAR-1990');

insert into member

values(member\_id\_seq.nextval,'LaDoris','Ngao','5 Mondrany','Bratislava','586-355-8882','08-MAR-1990');

insert into member

values(member\_id\_seq.nextval,'Midori','Nagayama','68 Via centrale','Sao Paolo','254-852-5764','17-JUN-1991');

insert into member

values(member\_id\_seq.nextval,'Mark','Quick\_to\_see','6921 King way','Lagos','63-559-7777','07-APR-1990');

insert into member

values(member\_id\_seq.nextval,'Audry','Ropeburn','86 Chu Street','Hong Kong','41-559-87','18-MAR-1991');

insert into member

values(member\_id\_seq.nextval,'Molly','Urguhart','3035 Laurier','Quebec','418-542-9988','18-JAN-1991');

**DATA INPUT-4(b):**

* insert into title

values(title\_id\_seq.nextval,'Willie and Christmas Too','All of Willies friends make a Christmas list for santa, but Willie is yet to add his own wish list.','G','CHILD','05-OCT-1995');

insert into title

values(title\_id\_seq.nextval,'Alien Again','Yet another installation of science fiction history. Can the heroine save the planet from the alien life from?','R','SCIFI','19-MAY-1995');

insert into title

values(title\_id\_seq.nextval,'The Glob','A meteor crashes near a small American town and unleashes carnvorous goo in this classic','NR','SCIFI','12-AUG-1995');

insert into title

values(title\_id\_seq.nextval,'The Day Off','With a little luck and a lot of ingenuity, a teenager skips school for a day in New York. ','PG','COMEDY','12-JUL-1995');

insert into title

values(title\_id\_seq.nextval,'Miracles on Ice','A six\_year\_old has doubts about santa Clause, but she discovers that miracles really do exist.','PG','DRAMA','12-SEP-1995');

insert into title

values(title\_id\_seq.nextval,'Soda Gang','After discovering a cache of drugs, a young couple find themselves pitted against a vicious gang.','NR','ACTION','10-JUN-1995');

**DATA INPUT-4(c):**

* insert into title\_copy

values(1,92,'AVAILABLE');

insert into title\_copy

values(1,93,'AVAILABLE');

insert into title\_copy

values(1,94,'AVAILABLE');

insert into title\_copy

values(1,95,'AVAILABLE');

insert into title\_copy

values(2,95,'AVAILABLE');

insert into title\_copy

values(3,95,'RENTED');

insert into title\_copy

values(1,96,'AVAILABLE');

insert into title\_copy

values(1,97,'AVAILABLE');

**DATA INPUT-4(d):**

* insert into rental(title\_id, copy\_id, member\_id, book\_date, exp\_ret\_date, act\_ret\_date)

values(92,1,101,sysdate-3,sysdate-1,null);

insert into rental(title\_id, copy\_id, member\_id, book\_date, exp\_ret\_date, act\_ret\_date)

values(93,2,101,sysdate-1,sysdate-1,null);

insert into rental(title\_id, copy\_id, member\_id, book\_date, exp\_ret\_date, act\_ret\_date)

values(95,3,102,sysdate-2,sysdate,null);

insert into rental(title\_id, copy\_id, member\_id, book\_date, exp\_ret\_date, act\_ret\_date)

values(97,1,106,sysdate-4,sysdate-2,null);

**PREVILAGE:**

* select\* from employees;( hr)
* select\* from hr.employees;(user)
* grant select on employees to Zihad;(hr)
* grant update (salary) on employees to Zihad;(hr)
* update hr.employees set salary =5000

where employee\_id=198;( user)

**Multitable**

* grant select on employees to Zihad;(hr)
* create table emp as select \* from hr.employees;
* select \* from emp;
* create table sal\_history

as select employee\_id,hire\_date, salary from emp;

create table mgr\_history

as select employee\_id,manager\_id,salary from emp;

* delete from sal\_history;

delete from mgr\_history;

* commit;
* insert all

into sal\_history values(empid,hiredate,sal)

into mgr\_history values(empid,mgr,sal)

select employee\_id empid,hire\_date hiredate,

salary sal, manager\_id mgr

from emp

where employee\_id>200;

* select \* from sal\_history;

select \* from mgr\_history;

* insert all

when sal>10000 then

into sal\_history values(empid,hiredate,sal)

when mgr >200 then

into mgr\_history values(empid,mgr,sal)

select employee\_id empid,hire\_date hiredate, salary sal, manager\_id mgr

from emp

where employee\_id>200;

* select \* from sal\_history;

select \* from mgr\_history;

**Rownum**

* select rownum, a.last\_name, a.salary

from(select last\_name, salary

from employees

order by salary desc) a

where rownum<4;

**Merge**

* create table product\_master(

id number(5) primary key,

name varchar2(20),

qty number(5)

);

* commit;
* insert into product\_master

values(101,'pp',10);

insert into product\_master

values(102,'xz',20);

insert into product\_master

values(103,'zz',50);

* create table product\_current(

id number(5) primary key,

name varchar2(20),

qty number(5)

);

* insert into product\_current

values(101,'pp',5);

insert into product\_current

values(102,'xz',-5);

insert into product\_current

values(103,'zz',10);

insert into product\_current

values(104,'abc',25);

* merge into product\_master pm

using product\_current pc

on (pm.id=pc.id)

when matched then

update set

pm.qty=pm.qty+pc.qty

when not matched then

insert values(pc.id,pc.name,pc.qty);

**Rollup**

* select department\_id,job\_id,sum(salary)

from employees

group by (department\_id,job\_id);

* select department\_id,job\_id,sum(salary)

from employees

group by rollup (department\_id,job\_id);

* select department\_id,job\_id,sum(salary)

from employees

group by cube(department\_id,job\_id);

* select\* from product\_current

**Session**

* select sessiontimezone,current\_date

from dual;

* alter session

set nls\_date\_format='dd\_mon\_yyyy hh24:mi:ss';

alter session set time\_zone='-5:0';

* select sessiontimezone,current\_date

from dual;

**Extract**

* select last\_name,hire\_date,extract(month from hire\_date)

from employees

where manager\_id=100;

**pairwise sub quary**

* select employee\_id,manager\_id,department\_id

from employees

where (manager\_id, department\_id)

in(select manager\_id,department\_id

from employees

where first\_name='John')

and first\_name<>'John';

**Correlated sub quary**

* select last\_name,salary,department\_id

from employees outer

where salary>(select avg(salary)

from employees

where department\_id=outer.department\_id);

**Exam**

* create table customer

(

customer\_id number(10),

customer\_name varchar2(20) not null,

age number(4),

course varchar2(15),

register\_date date

);

* insert into customer

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

* alter table customer

add email varchar2(50) unique;

* alter table customer

modify register\_date date default sysdate ;

* create sequence customer\_id\_seq

start with 100

increment by 10

maxvalue 10000

nocache

nocycle;

* insert into customer

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

* alter table customer

modify age number(4) constraint age\_ck check(age>=20);

* alter table customer

modify customer\_name varchar2(25) ;

* create table customer\_order(

order\_id number(10),

order\_date date,

product\_name varchar2(15),

price number(10,2),

quantity number(4),

customer\_id number(10)

);

* alter table customer\_order

add constraint order\_customer\_id\_fk foreign key(customer\_id) references customer(customer\_id);

* alter table customer

add constraint order\_customer\_id\_pk primary key(customer\_id);

* alter table customer\_order

add status varchar2(12);

* insert into customer\_order

values(1000,sysdate,'Cloth',50000,50,100,'purchase');

* insert into customer\_order

values(1100,sysdate,'Furniture',100000,25,110,'purchase');

* insert into customer\_order

values(1200,sysdate,'Cosmetic',60000,50,140,'purchase');

* select customer\_id, customer\_name, quantity\*price "total price"

from customer\_order natural join customer

* select customer\_name, sum(quantity\*price) "total price"

from customer\_order natural join customer

group by customer\_name;

* select customer\_name,product\_name,quantity\*price "total price"

from customer\_order natural join customer

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

* select product\_name,quantity,price

from customer\_order e

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

* update customer\_order

set product\_name='Pencil'

where order\_id in(12,14,16,24);

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

from customer\_order

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

**Exam Farid**

* create table emp1(

employee\_id number(3),

employee\_name varchar2(15),

employee\_department varchar2(4),

employee\_salary number(10),

employee\_email varchar2(20),

manager\_id number(4)

);

* alter table emp1

add constraint emp\_email\_uni unique(employee\_email);

* alter table emp1

add constraint emp\_dept\_pk primary key(employee\_department);

* create table dept

(

dept\_id number(4) not null,

dept\_name varchar2(10),

dept\_location varchar2(20),

dept\_phone number(20) unique

);

* alter table dept

add constraint dept\_name\_check\_ch check(dept\_name in('MIS','IT','SOFT','HR','ACC'));

* create sequence employee\_id\_seq

start with 10

increment by 1

maxvalue 1000

nocache

nocycle;

**Last class exam**

* create table customer\_sales(

order\_no number,

customer\_name varchar2(25) not null,

product\_name varchar2(20) not null,

price number(12,2),

quantity number(10),

order\_date date);

* create sequence order\_no\_seq

start with 1000

increment by 10

maxvalue 9999

nocache

nocycle;

* alter table customer\_sales

add tax number(3,2);

* alter table customer\_sales

modify order\_date default sysdate;

* alter table customer\_sales

modify product\_name varchar2(25);

* insert into customer\_sales

values(order\_no\_seq.nextval,'&customer\_name','&product\_name', &price,&quantity,sysdate,&tax)

* insert into customer\_sales

values(order\_no\_seq.nextval,'&customer\_name','&product\_name',&price,&quantity,sysdate-3,&tax)

* insert into customer\_sales

values(order\_no\_seq.nextval,'&customer\_name','&product\_name',&price,&quantity,sysdate-2,&tax)

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

from customer\_sales

group by order\_no;

* select product\_name, max(quantity),min(quantity)

from customer\_sales

group by product\_name;

* select product\_name,quantity

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;

* select order\_no,customer\_name,product\_name,quantity,price,tax, (quantity\*price)+(quantity\*price)\*tax total

from customer\_sales;

* select customer\_name,product\_name,sum(price)

from customer\_sales

group by rollup (customer\_name,product\_name);

* create view sales\_vu

as

select order\_no, customer\_name,product\_name,quantity

from customer\_sales;

* update customer\_sales

set price=(select avg(price) from customer\_sales)

where order\_no=1010;