Create / Insert / Update / Delete and Truncate rows - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.create table currency(id int primary key,code varchar(255),country varchar(255));

2.create table contact(id int primary key,org\_name varchar(255),street\_address1 varchar(255),street\_address2 varchar(255),city varchar(255),state varchar(255),postal\_code varchar(255),country\_code varchar(255),last\_name varchar(255),first\_name varchar(255),person\_title varchar(255),phone\_country\_code int,phone\_area\_code int,phone\_number varchar(255),email varchar(255),created\_time timestamp);

3.create table user\_details(id int primary key,username varchar(255),user\_password varchar(255),deleted int,created\_time timestamp,lastupdated\_time timestamp,lastlogin\_time timestamp,failed\_attempts int,contact\_id int REFERENCES contact(id)) ;

4.alter table contact modify country\_code int;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Create / Insert / Update / Delete and Truncate rows - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.insert  into credit\_card(id,card\_number,card\_expire,name,cc\_type) values (12, '454545214','211213','asd','asdd');

insert into credit\_card(id,card\_number,card\_expire,name,cc\_type) values (13, '4542145214','211713','asda','asdda');

insert into credit\_card(id,card\_number,card\_expire,name,cc\_type) values (42, '45454545214','218213','asdb','asddb');

2.insert into contact(id,org\_name ,street\_address1,street\_address2,city,state,postal\_code,country\_code,last\_name,first\_name,person\_title,phone\_country\_code,phone\_area\_code,phone\_number,email,created\_time) values (12, 'as','as','as','as','as','as','as','as','as','as',85,85,'as','as', to\_timestamp('2018-03-23 09:06:51.055000', 'YYYY-MM-DD HH24:MI:SS.FF6'));

 insert into contact(id,org\_name ,street\_address1,street\_address2,city,state,postal\_code,country\_code,last\_name,first\_name,person\_title,phone\_country\_code,phone\_area\_code,phone\_number,email,created\_time) values (13, 'asd','asd','asd','asd','asd','asd','asd','asd','asd','asd',86,86,'asd','asd',to\_timestamp('2018-02-23 09:06:51.055000', 'YYYY-MM-DD HH24:MI:SS.FF6'));

 insert into contact(id,org\_name ,street\_address1,street\_address2,city,state,postal\_code,country\_code,last\_name,first\_name,person\_title,phone\_country\_code,phone\_area\_code,phone\_number,email,created\_time) values (14, 'asf','asf','asf','asf','asf','asf','asf','asf','asf','asf',87,87,'asf','asf',to\_timestamp('2018-01-23 09:06:51.055000', 'YYYY-MM-DD HH24:MI:SS.FF6'));

3.UPDATE cheque

SET cheque\_number = '962541'

WHERE bank\_name = 'ICICI Bank';

4.

DELETE FROM cheque

WHERE bank\_name = 'Canara Bank';

5.insert into transaction\_type(id,type) values(12,'credit card');

insert into transaction\_type(id,type) values(13,'debit card');

insert into transaction\_type(id,type) values(14,'visa card');

6.UPDATE currency

SET code = 'USD'

WHERE country = 'United States';

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Create / Insert / Update / Delete and Truncate rows - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

create table address(id int primary key,work\_phone varchar(50),home\_phone varchar(50),fax\_phone varchar(50),email varchar(50),street varchar(50),city varchar(50),state varchar(50),zip\_code int,country varchar(50)) ;

create table package\_type(id int primary key,description varchar(30),package\_length number(\*,2),width number(\*,2),height number(\*,2),weight number(\*,2),volume number(\*,2));

alter table package\_type rename column description to notes;

alter table address rename column email to emailid;

create table shipment\_status(id int primary key,code varchar(30),status varchar(100),description varchar(100)) ;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

SQL\*Plus and SELECT Statement - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select \* from user\_details  order by id ;

2.select \* from invoice  order by id ;

3.select code,country from currency  order by code ;

4.select \* from credit\_card  order by name desc ;

5.select \* from transaction\_type  order by type  ;

6.select first\_name,last\_name from contact  order by first\_name  ;

7.select bank\_name,cheque\_date from cheque  order by bank\_name  ;

8.select first\_name,last\_name,org\_name,city,state from contact  order by first\_name  ;

9.select id,attempt,status,amount from payment  order by id desc  ;

10.select bill\_type,active\_since,active\_until from purchase\_order  order by active\_since ;

11.select username, created\_time, lastlogin\_time , failed\_attempts from user\_details  order by username ;

12.select processor, code, approval\_code , response\_msg from payment\_authorization  order by code ;

13.select  invoice\_id, type, amount, quantity , price from invoice\_line  order by invoice\_id ;

14.select   internal\_number, item name,deleted,price from item  order by internal\_number ;

15.select order\_id, item\_id, quantity, price,description from order\_line order by order\_id ;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

SQL\*Plus and SELECT Statement - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select \* from company  order by name ;

2.select id,description from rate where charge > 18000 order by id ;

3.select first\_name, middle\_name, last\_name, contact\_number from user\_details where role\_name = 'Customer' order by first\_name ;

4.select name,iata,fmc from agent order by name;

5.select id,street,email,home\_phone,zip\_code from address where city = 'Chennai' order by id;

6.select id,street,city,email,home\_phone from address where country = 'USA';

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Restrict / Sort / Customize Output - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select \* from credit\_card where cc\_type ='Visa'  order by name

2.select \* from cheque where bank\_name ='ICICI Bank'  order by cheque\_date;

3.select id,attempt,status,amount from payment where status ='Pending'  order by id;

4.select  id,due\_date,total,balance  from invoice where status ='Completed'  order by id;

5.select  \*  from user\_details where failed\_attempts >=2  order by id;

6.SELECT id,due\_date,total,balance  from invoice where EXTRACT(month FROM created\_time) in (3,4)

7.select  org\_name,first\_name,email,city  from contact where state= any('Karnataka','TamilNadu')  order by org\_name, first\_name;

8.select  username,user\_password  from user\_details where failed\_attempts >=1 order by username desc;

9.select name,internal\_number  from item where price >20000 order by name;

10.select id,notes  from customer where credit\_limit >20000 order by id;

11.select bill\_type,active\_since,active\_until from purchase\_order where status='Completed'  order by active\_since ;

12.select username, lastlogin\_time from user\_details where  extract(month from lastlogin\_time) =2 order by 2

13.select id,username from user\_details where deleted=0  order by id ;

14.select id, due\_date, total, status, balance,customer\_notes from invoice where payment\_attempts=any(0,1)  order by id ;

15.select code,country from currency where country=any('India','Singapore')  order by country ;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Restrict / Sort / Customize Output - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select first\_name, last\_name,contact\_number from user\_details where role\_name=any('Admin','Customer') order by first\_name;

2.select description,model from commodity\_identification where serial like ('%2') order by 2;

1.select id, name,identification\_number from shipment\_entity where credit\_limit > ANY(Select credit\_limit from shipment\_entity where name='Air Conditioner') order by 1;

3.select id,first\_name, last\_name,contact\_number,role\_name from user\_details where first\_name like ('S%') or first\_name like ('A%')  order by 1;

4.select id, description, cargo\_length, width, height, weight, volume from cargo where volume > ALL(200,300,350) order by 1;

1.select name from port where country='India' order by 1;

3.select id,first\_name, contact\_number,role\_name from user\_details where first\_name=some(select first\_name from user\_details where first\_name like ('S%') or first\_name like ('A%'))  order by 1

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Data Conversion & Aggregating Data - LC

select id, username,user\_password,deleted,created\_time,lastupdated\_time,lastlogin\_time,failed\_attempts,contact\_id from order by id;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select id, username,user\_password,deleted,created\_time,lastupdated\_time,lastlogin\_time,failed\_attempts,contact\_id from user\_details where EXTRACT(year FROM created\_time)=2015 order by id asc;

2.select distinct bank\_name from cheque order by bank\_name;

3.select max(credit\_limit) as max\_limit from customer;

4.select min(total) as min\_total from invoice;

5.select sum(balance) as total\_balance from invoice;

6.select count(id) as customer\_count from customer;

7.select avg(amount) as average\_payment from payment;

8.select id,period,bill\_type,active\_si,active\_un,created\_time,status,currency\_id,notes, created\_user\_id from purchase\_order where EXTRACT(year FROM created\_time)=2016 order by id asc;

9.select count(id) as Inactive\_Users from user\_details where lastlogin\_time is null  ;

10.select count(id) as Completed\_Orders from purchase\_order where status = 'Completed';

11.select first\_name,last\_name,phone\_number,email, nvl(last\_name,first\_name) updated\_last\_name from contact order by 1 desc

12.select paper\_invoice\_batch\_number, due\_date, last\_reminder, balance, CASE  WHEN balance >=25000 THEN 'Overdue' else 'Nil'  END  due\_status from invoice order by balance desc

13.select id, due\_date, total , balance from invoice where EXTRACT(month FROM created\_time) in (1,2,3) and EXTRACT(year FROM created\_time)=2015   order by balance desc

14.select invoice\_id, amount , created\_time from payment where EXTRACT(month FROM created\_time) in (5) and EXTRACT(year FROM created\_time)=2014   order by 1

15.select invoice\_id, amount  from payment where created\_time >=date '2015-01-01'  and created\_time <= date '2015-08-31' order by 2 desc

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Data Conversion & Aggregating Data - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

11.SELECT distinct extract ( year from DEPARTURE\_TIME) as year\_name, sum(TOTAL\_PROFIT)  OVER (PARTITION BY (extract ( year from DEPARTURE\_TIME)) )AS total\_profit\_by\_year  FROM   shipment order by year\_name;

3.Select notes from commodity where total\_value = (select min(total\_value) from commodity) order by 1;

4.Select carrier.carrier\_code,carrier.iata from carrier left join shipment\_entity on carrier.id=shipment\_entity.id where shipment\_entity.credit\_limit = (select max(shipment\_entity.credit\_limit) from shipment\_entity) ;

1.select  id, total\_weight, total\_volume from commodity where quantity=(select max(quantity) from commodity);

2.select role\_name, count(id) as user\_count from user\_details group by role\_name;

3.select count(status) as Shipment\_status\_count from shipment\_status

12.select \* from shipment where departure\_time = (select max(departure\_time) from shipment);

7.select address.city, count(agent.address\_id) as Agent\_Count from address left join agent on address.id=agent.address\_id where agent.address\_id is not null group by address.city order by address.city;

13.select description,package\_length,width,height from package\_type where volume=(select max(volume) from package\_type);

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Joins and Subqueries - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select customer.\* from customer left join user\_details on customer.created\_user\_id=user\_details.id where user\_details.username='arun' order by customer.id;

2.select a.org\_name, a.street\_address1, a.street\_address2, a.city, a.state, a.phone\_number,a.email from contact a where a.id in (select b.contact\_id from customer b where b.credit\_limit>25000) order by 1

3.select a.first\_name, a.phone\_number, a.email from contact a left join user\_details b on a.id=b.contact\_id where b.lastlogin\_time is null order by 1;

6.elect paper\_invoice\_batch\_number,due\_date,last\_reminder, balance from invoice join customer on customer.id= invoice.customer\_id

join contact on contact.id=customer.contact\_id where org\_name='Cognizant' order by balance desc, last\_reminder asc

7.select bill\_type, active\_since, active\_until, status from purchase\_order join user\_details on purchase\_order.created\_user\_id=user\_details.id

where user\_details.contact\_id=(select contact\_id from user\_details join contact on user\_details.contact\_id=contact.id where contact.first\_name='Arun') order by bill\_type asc;

8.select invoice\_id, amount, status from payment where invoice\_id=ANY(select id from invoice where customer\_id=ANY(select id from customer where contact\_id = ANY(select id from contact where org\_name='ABC Tech Park')))

order by amount desc;

13.SELECT \*

FROM (SELECT p.amount as amount,type

FROM payment p

Join transaction\_type t

on(p.transaction\_type\_id = t.id))

PIVOT (SUM(amount) FOR (type) IN ('Credit Card' AS total\_credit, 'Cheque' AS total\_cheque))

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Joins and Subqueries - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select c.id,c.iata,c.carrier\_code from carrier c left join address a on c.address\_id=a.id where a.country='India' order by c.iata;

2.select ag.\* from agent ag  left join address ad on ag.address\_id=ad.id where ad.state = any('Karnataka','TamilNadu') order by ag.iata;

3.select c.description,c.cargo\_length,c.width,c.weight,s.name,s.booking\_number,s.total\_profit from shipment s, cargo c ,shipment\_cargo sc

where sc.cargo\_id= c.id

and s.id= sc.shipment\_id

and  c.volume>300 order by s.total\_profit desc;

4.select a.\* from shipment\_entity a, shipment b ,port c

where c.id= b.arrival\_port\_id

and a.id= b.shipment\_entity\_id

and  c.name='Mumbai' order by a.id;select a.\* from commodity\_identification a left join rate b on a.rate\_id=b.id where b.charge = (select max(b.charge) from rate b) order by  a.description;

5.select a.\* from commodity\_identification a left join rate b on a.rate\_id=b.id where b.charge = (select max(b.charge) from rate b) order by  a.description;

6.select a.\* from shipment\_entity a, shipment b ,port c

where c.id= b.arrival\_port\_id

and a.id= b.shipment\_entity\_id

and  c.name=any('Chennai','Cochin') order by a.id;

7.select a.name,a.iata,a.fmc,b.work\_phone,b.email,b.city,b.state,b.country from company a,address b where a.address\_id=b.id order by 1 desc;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

SET Operators and DDL - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select p.id,p.amount,p.status from payment p left join transaction\_type t on p.transaction\_type\_id=t.id

where t.type='Credit Card'

union

select p.id,p.amount,p.status from payment p left join transaction\_type t on p.transaction\_type\_id=t.id

where t.type='Cheque' order by 1;

2.select count(p.id) Order\_count from purchase\_order p left join user\_details u on p.created\_user\_id=u.id where u.username='Nishasree'

Union

select count(p.id) Order\_count from purchase\_order p left join user\_details u on p.created\_user\_id=u.id where u.username='Shivani';

3.select a.id,a.name from item a

inner join invoice\_line b on a.id =b.item\_id

inner join invoice e on b.invoice\_id=e.id

inner join customer f on e.customer\_id=f.id

inner join contact g on f.contact\_id=g.id

WHERE g.org\_name = 'Cognizant'

UNION ALL

select a.id,a.name from item a

inner join invoice\_line b on a.id =b.item\_id

inner join invoice e on b.invoice\_id=e.id

inner join customer f on e.customer\_id=f.id

inner join contact g on f.contact\_id=g.id

WHERE g.org\_name = 'ABC Tech Park'

order by 1,2

4.SELECT item.id, item.name FROM item INNER JOIN invoice\_line ON item.id = invoice\_line.item\_id

INNER JOIN invoice ON invoice\_line.invoice\_id = invoice.id

INNER JOIN customer ON customer.id = invoice.customer\_id

INNER JOIN contact ON customer.contact\_id = contact.id

WHERE contact.org\_name = 'Cognizant'

union

SELECT item.id, item.name FROM item

INNER JOIN invoice\_line ON item.id = invoice\_line.item\_id

INNER JOIN invoice ON invoice\_line.invoice\_id = invoice.id

INNER JOIN customer ON customer.id = invoice.customer\_id

INNER JOIN contact ON customer.contact\_id = contact.id

WHERE contact.org\_name = 'ABC Tech Park';

5.SELECT iv.id, op.order\_id, count(ol.quantity) AS item\_count

FROM invoice iv

JOIN order\_process op ON iv.id = op.invoice\_id

JOIN purchase\_order po ON op.order\_id = po.id

JOIN order\_line ol ON ol.order\_id = po.id

JOIN item i ON ol.item\_id = i.id

where iv.status = 'Completed'

GROUP BY iv.id, op.order\_id

UNION

SELECT iv.id, op.order\_id, count(ol.quantity) AS item\_count

FROM invoice iv

JOIN order\_process op ON iv.id = op.invoice\_id

JOIN purchase\_order po ON op.order\_id = po.id

JOIN order\_line ol ON ol.order\_id = po.id

JOIN item i ON ol.item\_id = i.id

where iv.status = 'Pending'

GROUP BY iv.id, op.order\_id

ORDER BY 1

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

SET Operators and DDL - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select name from port where id in(

select departure\_port\_id from shipment having count(departure\_port\_id) =(

select max(a) from (select count(departure\_port\_id)as a from shipment group by departure\_port\_id)) group by departure\_port\_id)

union

select name from port where id in(

select departure\_port\_id from shipment having count(departure\_port\_id) =(

select min(a) from (select count(departure\_port\_id)as a from shipment group by departure\_port\_id)) group by departure\_port\_id)

order by 1;

2.SELECT SE.NAME, TOTAL\_WEIGHT, TOTAL\_VOLUME, TOTAL\_PROFIT FROM SHIPMENT SE

INNER JOIN SHIPMENT\_CARGO S ON SE.ID = S.SHIPMENT\_ID

INNER JOIN CARGO C ON S.CARGO\_ID = C.ID

INNER JOIN PACKAGE\_TYPE M ON C.PACKAGE\_TYPE\_ID = M.ID

WHERE M.DESCRIPTION = 'Pallet'

intersect

SELECT SE.NAME, TOTAL\_WEIGHT, TOTAL\_VOLUME, TOTAL\_PROFIT FROM SHIPMENT SE

INNER JOIN SHIPMENT\_CARGO S ON SE.ID = S.SHIPMENT\_ID

INNER JOIN CARGO C ON S.CARGO\_ID = C.ID

INNER JOIN PACKAGE\_TYPE M ON C.PACKAGE\_TYPE\_ID = M.ID

WHERE M.DESCRIPTION = 'Carton'

3.select se.id, se.name from shipment\_entity se

inner join shipment s on se.id = s.shipment\_entity\_id

inner join port p on s.departure\_port\_id = p.id

inner join method m on s.method\_id=m.id

where m.name = 'DHL' AND p.name = 'Mumbai'

intersect

select se.id, se.name from shipment\_entity se

inner join shipment s on se.id = s.shipment\_entity\_id

inner join port p on s.departure\_port\_id = p.id

inner join method m on s.method\_id=m.id

where m.name = 'TNT' AND p.name = 'Mumbai'

4.select \* from

(select c.id, i.model, i.serial, i.invoice\_number from commodity c

inner join cargo ca on ca.id=c.cargo\_id

inner join commodity\_identification i on i.id=c.commodity\_identification\_id

where ca.volume>300

union all

select c.id, i.model, i.serial, i.invoice\_number from commodity c

inner join cargo ca on ca.id=c.cargo\_id

inner join commodity\_identification i on i.id=c.commodity\_identification\_id

inner join rate r on r.id=i.rate\_id

where r.charge>18000)

order by id

5.select b.id, b.model, b.serial,b.invoice\_number from rate a

inner join commodity\_identification b on a.id = b.id

inner join commodity c on b.id=c.commodity\_identification\_id

inner join cargo d on c.cargo\_id = d.id

where d.volume > 300

minus

select b.id, b.model, b.serial,b.invoice\_number from rate a

inner join commodity\_identification b on a.id = b.id

inner join commodity c on b.id=c.commodity\_identification\_id

inner join cargo d on c.cargo\_id = d.id

where a.charge < 18000

order by id asc

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Views / Synonyms / Triggers - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.create synonym client for customer

2.create synonym product for item

3.create synonym money for currency

4.create view transaction\_type\_details as select \* from transaction\_type

5.create view invoice\_details as select \* from invoice where balance=0

6.create view cheque\_details as select \* from cheque where bank\_name='Yes Bank'

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Views / Synonyms / Triggers - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.create view shipment\_details as select shipment.\* from shipment left join package\_type on shipment.id=package\_type.id where  package\_type.description='Carton';

2.create view address\_details as select \* from address where state='TamilNadu';

3.create view user\_info as select u.\* from user\_details u left join address a on u.address\_id=a.id where a.email like ('%yahoo%');

4.create view company\_details as select \* from company;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Strings and Regular Expressions - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.SELECT CONCAT(CONCAT(first\_name, ' '),last\_name) full\_name,city, state, phone\_number,email from contact where state='Karnataka' order by 2,1;

2.select id,notes from customer where notes like ('%Recruit%') order by id;

3.select username,rawtohex(user\_password) encrypted\_password,lastlogin\_time , failed\_attempts from user\_details order by 1;

4.select username,rawtohex(user\_password) encrypted\_password,lastlogin\_time , failed\_attempts from user\_details where length(rawtohex(user\_password))>15  order by 1;

5.SELECT RPAD(card\_number, 20, '#') modified\_number FROM credit\_card order by 1;

6.SELECT LPAD(transaction\_id, 30, '#') modified\_id FROM payment\_authorization order by 1;

7.SELECT REPLACE(username, 'a','\*') modified\_name FROM user\_details order by 1;

8.SELECT SUBSTR(code,0 , 2) modified\_code FROM currency order by 1;

13.select first\_name from contact where length(first\_name)<=5 order by 1;

select nr, regexp\_replace(nr,'(^[[:digit:]]{3})([[:digit:]]{2})([[:digit:]]{4}$)','\1-\2-\3') formatted\_nr from t

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Strings and Regular Expressions - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

12.select count(id) agent\_count from agent where name like ('I%');

7.select distinct(SUBSTR(state,0 , 3)) state\_code from address order by 1;

8.select u.first\_name,u.last\_name,a.email from user\_details u, address a where a.id=u.address\_id and length(u.user\_password)<9 order by 1;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Analytical Functions - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.select id, invoice\_id, amount, AVG(amount) OVER (PARTITION BY invoice\_id) as average\_payment from payment order by invoice\_id;

2.select id, invoice\_id, amount,AVG(amount) OVER (PARTITION BY invoice\_id ORDER BY amount ASC )as average\_payment from payment

4.select invoice\_id, amount, round(cume\_dist() over (partition by invoice\_id order by amount) \* 100,2) as cume\_dist\_payment from payment

5.select customer\_id, total, row\_number() over (partition by customer\_id ORDER BY id ) SNO from invoice order by customer\_id, sno asc

6.select distinct a.customer\_id

,b.customer\_total

,dense\_rank() over ( order by b.customer\_total desc) as rank

from invoice a

join ( select customer\_id

,sum(total) as customer\_total

from invoice

group by customer\_id) b

on a.customer\_id = b.customer\_id

order by rank

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Analytical Functions - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

8.SELECT port.name,departure\_time,total\_volume,

SUM(total\_volume) OVER () AS Total\_sum\_volume,

Avg(total\_volume) OVER (PARTITION BY departure\_port\_id) AS "Average\_volume\_By\_port",

SUM(total\_volume) OVER (PARTITION BY shipment.id ORDER BY departure\_time) AS "Total\_volume\_by\_date"

FROM shipment inner join port on port.id =shipment.departure\_port\_id

order by port.name;

7.select id,name,departure\_time,LEAD (departure\_time,1) OVER (ORDER BY id) as Next\_shipment\_date from shipment

5.select port.name, total\_weight, min(total\_weight) over (partition by port.name) as port\_min\_weight

 from shipment inner join port on port.id=shipment.departure\_port\_id order by 1;

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

PL/SQL datatypes and programming constructs - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.create or replace procedure insert\_credit( credit\_id in number,card\_number in varchar,card\_expire in varchar,name in varchar,cc\_type in varchar) as

begin

insert into credit\_card(id,card\_number,card\_expire,name,cc\_type) values (credit\_id,card\_number,card\_expire,name,cc\_type);

end;

/

2.create or replace procedure payment\_count(

p\_count out Number

)

IS

Begin

SELECT count(\*) as successful\_payment\_count INTO p\_count

FROM payment\_authorization WHERE response\_msg = 'Successful';

End;

/

3.create or replace function find\_credit\_card(card\_no IN varchar)

return varchar

is holder\_name varchar(32767);

p\_holder\_name varchar(32767);

BEGIN

select name into p\_holder\_name from credit\_card where card\_number = card\_no;

holder\_name:=p\_holder\_name;

return holder\_name;

EXCEPTION

when NO\_DATA\_FOUND then

holder\_name:='No such Type';

return holder\_name;

END;

/

4.CREATE OR REPLACE FUNCTION find\_balance (invoice\_id IN INTEGER)

RETURN VARCHAR

IS

TEMP NUMBER;

balance\_status VARCHAR(255);

BEGIN

SELECT balance

INTO TEMP

FROM invoice

WHERE id = invoice\_id;

IF TEMP = 0 THEN

balance\_status := 'The payment has been Completed';

ELSE

balance\_status :=TEMP ||' '|| 'yet to be paid';

END IF;

RETURN balance\_status;

END find\_balance;

/

5.create or replace procedure select\_city (user\_id In Number, city\_details Out varchar) IS

l\_city\_name contact.city%TYPE;

BEGIN

select co.city

INTO l\_city\_name

from user\_details ud

INNER JOIN contact co ON ud.contact\_id = co.id

where ud.id = user\_id;

IF l\_city\_name = 'Bangalore' Then

city\_details := 'User is from Bangalore';

ELSIF l\_city\_name = 'Chennai' Then

city\_details := 'User is from Chennai';

ELSE

city\_details := 'User is from other cities';

END IF;

end select\_city;

/

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

PL/SQL datatypes and programming constructs - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.create or replace function credit\_limit (entity\_id IN INTEGER)

RETURN VARCHAR

IS

c\_credit\_limit NUMBER;

limit\_status VARCHAR(255);

BEGIN

select credit\_limit

INTO c\_credit\_limit

from shipment\_entity

where id = entity\_id;

IF c\_credit\_limit > 50000 THEN

limit\_status := 'Credit limit is greater than 50000';

ELSE

limit\_status :='Credit limit is less than 50000';

END IF;

RETURN limit\_status;

END credit\_limit;

/

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Cursors and Exception Handling - LC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

1.create or replace procedure select\_invoice\_line(invoice\_line\_id in number,invoice\_line\_details out SYS\_REFCURSOR )

as

begin

open invoice\_line\_details for

select type,amount, quantity from invoice\_line where id=invoice\_line\_id;

end;

/

2.create or replace procedure select\_invoice(

invoice\_id IN invoice.id%TYPE,

invoice\_details OUT SYS\_REFCURSOR

) AS

BEGIN

OPEN invoice\_details For

select status, paper\_invoice\_batch\_number

from invoice

where id =invoice\_id;

END select\_invoice;

/

3.CREATE OR REPLACE FUNCTION find\_transaction\_type (transaction\_type\_id INT )

RETURN VARCHAR

IS

trans\_type VARCHAR(255);

BEGIN

SELECT type

INTO trans\_type

FROM transaction\_type

WHERE id = transaction\_type\_id;

RETURN trans\_type;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'No Such Type';

END;

/

4.CREATE OR REPLACE FUNCTION find\_invoice\_status (cust\_id INT )

RETURN VARCHAR

IS

invoice\_status VARCHAR(255);

BEGIN

SELECT status

INTO invoice\_status

FROM invoice

WHERE customer\_id = cust\_id;

RETURN invoice\_status;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'No Such Customer';

WHEN TOO\_MANY\_ROWS THEN

RETURN 'Multiple Rows Returned';

END;

/

5.create or replace FUNCTION find\_authorization

(trans\_id IN varchar)

RETURN varchar

IS

proc\_name varchar(150);

BEGIN

select processor into proc\_name from payment\_authorization where transaction\_id = trans\_id ;

IF SQL%NOTFOUND THEN

proc\_name := 'No Such Payment Authorization';

RETURN proc\_name;

ELSIF SQL%FOUND THEN

RETURN proc\_name;

END IF;

END;

/

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

Cursors and Exception Handling - CC

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

create or replace procedure find\_shipment (

status\_id IN shipment.shipment\_status\_id%TYPE,

shipment\_details OUT SYS\_REFCURSOR

) AS

BEGIN

OPEN shipment\_details FOR

select id, name, booking\_number, departure\_time ,arrival\_time

from shipment

where id = status\_id order by status\_id asc;

end find\_shipment;

/

- - - - - - - - - - - - - - - - - - - - - - - - - - - -  -

CREATE OR REPLACE FUNCTION find\_port (p\_id IN INTEGER)

RETURN number

IS

transportation\_count number;

TEMP Varchar(255);

BEGIN

SELECT  is\_rail INTO TEMP FROM port WHERE  id=p\_id ;

IF TEMP = 'true' THEN

transportation\_count :=1;

ELSE

transportation\_count :=0;

END IF;

RETURN transportation\_count;

END find\_port;

/

-----------------------------------------------------------------------

create or replace procedure company\_count(

c\_count out Number

)

IS

begin

SELECT count(\*) as count\_of\_companies into c\_count FROM Company c, Address a WHERE c.address\_id=a.id and a.country='USA';

dbms\_output.put\_line('count\_of\_companies' ||c\_count);

End;

/