**SQL ASSIGNMENTS**

**Assignment 1**

1) create table member

(

Member\_Id number(5),

Member\_Name varchar2(30),

Member\_address varchar2(50),

Acc\_Open\_Date date,

Membership\_type varchar2(20),

Fees\_paid number(4),

Max\_Books\_Allowed number(2),

Penalty\_Amount number(7,2)

);

create table books

(

book\_no number(6),

book\_name varchar2(30),

author\_name varchar2(30),

cost number(7,2),

category char(10)

);

create table issue

(Lib\_Issue\_id Number(10),

Book\_No number(6) ,

Member\_Id number(5) ,

Issue\_Date date,

Return\_date date,

);

2) desc member

3)drop table member

4) create table member

(

Member\_Id number(5) PRIMARY KEY,

Member\_Name varchar2(30),

Member\_address varchar2(50),

Acc\_Open\_Date date,

Membership\_type varchar2(20),

Fees\_paid number(4),

Max\_Books\_Allowed number(2),

Penalty\_Amount number(7,2)

);

create table member

(

Member\_Id number(5) PRIMARY KEY,

Membership\_type varchar2(20),

Member\_Name varchar2(20)

);

5) ALTER TABLE member MODIFY Member\_Name varchar2(30);

6) Alter table Issue ADD Referenceof Char(30);

7) Alter table Issue DROP column Referenceof ;

8) Alter table Issue RENAME TO Lib\_Issue;

9) insert into member values(1,'Richa Sharma','Pune','10-DEC-2005','lIFETIME',25000,5,50);

insert into member values(2,'Garima Sen','Pune','17-OCT-2018','ANNUAL',1000,3,'');

10) insert into member values(3,'Riya','Mumbai','11-OCT-2018','Annual',1000,24,300);

insert into member values(4,'Piya','Jogeshwari','13-OCT-2018','quarterly',500,12,30);

insert into member values(5,'Jiya','ghansoli','12-OCT-2018','Half Yearly',600,20,150);

11) ALTER TABLE member MODIFY Member\_Name nvarchar2(20);

12) insert into member values(6,'Diya','ghansoli','12-OCT-2018','Half Yearly',600,110,150);

NEED TO SPECIFY REASON

QL Error: ORA-01438: value larger than specified precision allowed for this column01438. 00000 - "value larger than specified precision allowed for this column"

When inserting or updating records, a numeric value was enteredthat exceeded the precision defined for the column

13) create table Member101

(

Member\_id number(5)PRIMARY KEY,

Member\_Name varchar2 (30),

Member\_address varchar2 (50),

Acc\_Open\_Date date,

Membership\_type varchar2 (20),

Fees\_Paid Number(4),

Max\_Books\_Allowed Number(2),

Penalty\_Amount Number(7,2)

);

14) a) Alter table Member101 ADD CONSTRAINT Member101 CHECK(Max\_Books\_Allowed < 100);

b) Alter table Member101 ADD CONSTRAINT Mem101 CHECK(Penalty\_Amount <=1000);

15) drop table books;

16) create table books

(

book\_no number(6) primary key,

book\_name varchar2(30) not null,

author\_name varchar2(30),

cost number(7,2),

category char(10)

);

17) insert into books values(101,'LET US C','DENNIS RITCHIE',450,'SYSTEM');

insert into books values(102,'ORACLE-COMPLETE REF','LONI',550,'DATABASE');

insert into books values(103,'MASTERING SQL','LONI',250,'DATABASE');

insert into books values(104,'PL SQL-REF','SCOTT URMAN',750,'DATABASE');

18) insert into Books values(1111,'&Mastering SQL','&Loni',250, '&Database');

19) create table book101

(

book\_no number(6) primary key,

book\_name varchar2(30) not null,

author\_name varchar2(30),

cost number(7,2),

category char(10)

);

20) INSERT INTO Books101

SELECT \* FROM books;

21) commit; [oracle autocommited]

22) select \* from books;

select \* from book101;

23) insert into books values(105,'NATIONAL GEOGRAPHIC','ADIS SCOTT',1000,'SCIENCE');

24) rollback;

25) update books set cost = 300, category = 'RDBMS' where book\_no = 103;

26) Alter table Lib\_Issue RENAME TO Issue;

27) drop table issue

28.create table Issue

(Lib\_Issue\_id Number(10), CONSTRAINT PK Primary Key(Lib\_Issue\_id) ,

Book\_No number(6) REFERENCES Books(Book\_no),

Member\_Id number(5) references Member(Member\_Id),

Issue\_Date date,

Return\_date date,

CONSTRAINT date\_CHECK CHECK (Issue\_date<Return\_date));

29.insert into issue values(7006,101,3,'18-Feb-2006',null)

insert into issue values(7005,104,2,'15-Nov-2006',null)

insert into issue values(7004,101,1,'04-Jul-2006',null)

insert into issue values(7003,104,1,'15-Jan-2006',null)

insert into issue values(7002,102,2,'25-Dec-2006',null)

30. save the data-COMMIT

31. ALTER TABLE ISSUE DISABLE CONSTRAINT date\_CHECK;

ALTER TABLE ISSUE DISABLE CONSTRAINT PK;

32. ORA-02298: cannot validate (SYSTEM.FK\_MEMBER\_ID) - parent keys not found////ERROR

33. ALTER TABLE ISSUE ENABLE CONSTRAINT DATE\_CHECK ;

34. delete from issue where member\_id=3

alter table issue enable constraint fk\_member\_id;

35. Delete from member where member\_id=1;

Error :integrity error child record found constraint violated

36. UPDATE ISSUE

SET Return\_Date = Issue\_Date + 15

WHERE Lib\_Issue\_Id IN (7004, 7005);

37. UPDATE MEMBER

SET Penalty\_Amount=100 WHERE MEMBER\_NAME='Garima Sen';

38. Savepoint X;

39 . delete from Issue where (Member\_id=1 AND Issue\_date <= '10-Dec-2006');

40. DELETE From Books WHERE (Category!= 'RDBMS' AND Category!='DATABASE' );

41. Rollback to SAVEPOINT X;

42. commit;

43. drop table Member101;

44. drop table Book101;

45. desc books;

desc member;

desc issue;

46.create sequence no\_seq

start with 100

increment by 2

maxvalue 200

nocycle;

47. drop sequence no\_seq

48. create sequence book\_seq

start with 101

increment by 1

maxvalue 1000

nocycle;

49) create sequence member\_seq

start with 1

increment by 1

maxvalue 100

nocycle;

50. drop sequence member\_seq;

drop sequence book\_seq;

**Assignment 2**

1. select book\_name from books where author\_name = 'loni' and cost < 600;
2. select \* from issue where return\_date > CURRENT\_DATE;
3. update issue set return\_date = '31-dec-2006' where return\_date = NULL and member\_id not in(7005,7006);
4. select \* from issue where (to\_date(return\_date)-to\_date(issue\_date))>30;
5. select \* from books where cost in (500,750) and category = 'Database';
6. . select \* from books where category in ('Science','Database','Fiction','Management');

7.select \* from members order by penalty\_amount desc;

8. select \* from books order by category, price desc;

9. select \* from books where book\_name like '%SQL%';

10. select \* from member where ((member\_name like 'R%' or member\_name like 'G%')and(member\_name like '%I%' or member\_name like '%I'));

11. select i.book\_no, b.book\_name, i.issue\_date from issue i, books b where i.book\_no = b.book\_no AND extract(month from issue\_date) = 12 AND b.category = 'Database';

12. create view book\_count\_permember as select member\_id, count(member\_id) as no\_of\_books\_issued from issue group by member\_id order by count(member\_id) desc select m.member\_id, m.member\_name, b.no\_of\_books\_issued from member m, book\_count\_permember b where b.member\_id = m.member\_id order by b.no\_of\_books\_issued descFROM member m, issue i ;

13. select i.member\_id, m.member\_name, b.book\_no, b.book\_name from issue i, books b, member m where m.member\_id = i.member\_id AND i.book\_no = b.book\_no AND m.member\_id = 1 ;

14. select member\_name from member where acc\_open\_date between '01-jan-2006' and '01-jan-2007';

15. select lib\_issue\_id, issue\_date, return\_date, (to\_date(return\_date)-to\_date(issue\_date)) as days\_issued from issue;

16. select \* from member order by acc\_open\_date;

17. select count(lib\_issue\_id) from issue where member\_id = 101;

18. select sum(penalty\_amount) from member;

19. select count(member\_id) from member;

20. select count(lib\_issue\_id) from issue;

21. select avg(fees\_paid) from member;

22. select months\_between(to\_date(return\_date),to\_date(issue\_date)) from issue;

23. select length (member\_name) FROM member ;

24. select substr(membership\_type,1,5) as membership\_type from member;

25. select last\_day(issue\_date) from issue;

**Assigment 3:**

1.select category, count(category) from books group by category;

2.select book\_no, count(book\_no) from issue group by book\_no order by 2 desc;

3.select max(penalty\_amount) "Maximum penalty",min(penalty\_amount) "Minimum penalty",sum(penalty\_amount) "total penalty",round(avg(penalty\_amount)) "Average penalty" from member;

4.select member\_id,count(book\_no) from issue group by member\_id having count(book\_no)>2;

5.select book\_no,count(book\_no),member\_id from issue group by book\_no order by 2 desc;

6.select to\_char(issue\_date,'mm') , count(book\_no) from issue group by issue\_date order by issue\_date desc;

7.select book\_no from issue where issue\_date=null;

8.select distinct(member\_id) from member where member\_id in (select member\_id from issue);

9. select member\_id, count(member\_id) from issue group by member\_id having count(member\_id) in ((select max(count(member\_id)) from issue group by member\_id),(select min(count(member\_id)) from issue group by member\_id));

10. SELECT \* FROM issue WHERE extract(month FROM issue\_date) IN (07,12)

11.select b.book\_no,b.book\_name,s.issue\_date from books b INNER JOIN issue s ON to\_char(s.issue\_date,'mon')='DEC' and b.category='DATABASE';

12. Select m.member\_id,m.member\_name,count(i.book\_no)from member m,issue i

Where m.member\_id=i.member\_id group by m.member\_id,m.member\_name order by count(i.book\_no) desc;

13. select i.member\_id, m.member\_name, b.book\_no, b.book\_name from issue i, books b, member m where m.member\_id = i.member\_id AND i.book\_no = b.book\_no AND m.member\_id = 1 ;

14.select \* from Member where Member\_id in (select Member\_id from Issue WHERE Book\_no=(SELECT BOOK\_NO from Books WHERE Category='database'));

15. select Book\_NAME,Cost From Books where Cost IN(SELECT max(Cost) from Books Group by Category);

16) select m.member\_name, i.issue\_date, m.acc\_open\_date, i.return\_date from member m, issue i where i.issue\_date not between m.acc\_open\_date and i.return\_date;

17) select member\_id, member\_name from member where member\_id not in (select member\_id from issue);

18. select \* from member where Max\_Books\_Allowed <(select count (Book\_No) from Issue);

19) SELECT distinct o.member\_id, n.member\_name, o.book\_no from issue o, issue g, member n where o.book\_no in (select book\_no from issue where member\_id = 2) and o.book\_no = g.book\_no and o.member\_id <> 2 and n.member\_id = o.member\_id;

20) select book\_name, book\_no from books where book\_no in (select book\_no from issue where (SYSDATE-issue\_date)>30);

21) select book\_name, author\_name from books where author\_name in (select author\_name from books group by author\_name having count(author\_name)>1);

22) select \* from book\_count where count\_books in ((SELECT max(count\_books) from book\_count),(SELECT min(count\_books) from book\_count));

23) Select \* from books where rownum <= 3 order by cost desc;

24) select sum(cost) from books where book\_no in (select book\_no from issue where return\_date>sysdate );

25) create view max\_issue\_book as select book\_no, count(book\_no) as no\_of\_issue from issue group by book\_no having count(book\_no) = (select max(count(book\_no)) from issue group by book\_no);

select m.book\_no, b.book\_name, b.author\_name, b.cost, b.category from max\_issue\_book m, books b where m.book\_no = b.book\_no;

26) SELECT count(Book\_No)from Issue where member\_id in (select member\_id from member where Membership\_type='lifetime');

27) select membership\_type, count(membership\_type)"count" from member group by membership\_type;

28)Create view issue\_count as select member\_id, count(member\_id) as books\_issued from issue group by member\_id order by count(member\_id) desc;

select i.member\_id, m.member\_name, m.membership\_type, i.books\_issued from issue\_count i, member m where m.member\_id = i.member\_id AND rownum<=5

29)create view issue\_count as select member\_id, count(member\_id) as books\_issued from issue group by member\_id order by count(member\_id) desc;

select i.member\_id, m.member\_name, m.membership\_type, i.books\_issued from issue\_count i, member m where m.member\_id = i.member\_id AND rownum<=3

30) select member\_id, member\_name from member where rownum <= 5 order by acc\_open\_date;

31.select m.member\_name,m.membership\_type,s.issue\_date from member m, issue s where to\_char(issue\_date,'dd-mon-yy') between '1-dec-06' and '31-dec-06' ;

32) select member\_id, member\_name from member where member\_id in (select member\_id from issue where sysdate <= return\_date);

33.select member\_name from member where acc\_open\_date=(select acc\_open\_date from member where member\_name='Garima Sen' ) ;

34) select MEMBER\_NAME from Member where Member\_id in(select i.member\_id from issue i INNER JOIN books b ON to\_char(issue\_date,'mm')=12 and b.author\_name='loni');

35 ) SELECT \* from(select b.author\_name from

Books b,Member m,Issue i where b.book\_no=i.book\_no and m.member\_id=i.member\_id AND m.membership\_type='LIFETIME' ORDER by i.issue\_date);

36) SELECT \* from (Select b.author\_name from

Books b,member m,issue i where b.book\_no=i.book\_no and m.member\_id=i.member\_id AND m.membership\_type='Annual')where rownum<=1;

37) SELECT \* from(Select \* from member m where membership\_type='Annual')where rownum<=5;

38) select member\_id from issue where book\_no=(select book\_no from Books where cost>300 and author\_name='loni');

SELECT MEMBER\_NAME FROM MEMBER WHERE MEMBERSHIP\_TYPE= 'Yearly' AND (ACC\_OPEN\_DATE BETWEEN '01-jAN-2006' AND '31-dEC-2006') where book\_no in(select i.book\_no from Issue i, member m where i.member\_id=m.member\_id group by book\_no having count(book\_no)=1);

39) create view member\_id\_type as select membership\_type, member\_id from member group by membership\_type, member\_id; create view temp as select m.membership\_type, m.member\_id, i.book\_no, b.category from issue i, member\_id\_type m, books b where i.member\_id = m.member\_id AND i.book\_no = b.book\_no; select category, membership\_type, count(category) as count from temp group by category, membership\_type

40) select member\_name, membership\_type from member where member\_id in(select member\_id from issue) and membership\_type='Lifetime' and acc\_open\_date between '01-jan-2006' and '31-Dec-2006'