

# DBMS

## ① Library

a) Select b.id, b.title, b.pubname, a.authname, c.noofcopies, d.bname from book b, bookauth a, bookcopies c, libbranch d where b.bookid = a.bookid AND . . .

b) Select ~~a.dno~~ from ~~d.dno~~ where bookid ~~&~~ 6 bookcopies where date out between '01-JAN-2017' AND '01-JUL-2017' ~~AND~~ GROUP BY ~~d.dno~~ ~~where~~ ~~count~~ having count(\*) > 3;

c) Delete from book where bookid = 3;

~~delete~~ d) ~~select view~~ CREATE VIEW V\_PUB AS select PubYear from Book;

e) CREATE VIEW VBOOKS AS select b.bookid, b.title, c.noofcopies from book b, bookcopies c, libbranch d where c.bookid = b.bookid AND c.branchid = d.branchid;

## ② Sales order

a) Select ~~custid~~ from customer where grade > (select avg(grade) from customer where city = 'baj');

my  
1st cut.



~~IK 18~~

② ~~Each~~ taxman  $s$ , customer  $c$  where  $s.city = c.city$

d) little doubt ✓ { salesman where city not in (select city from customer);

distinct

~~(V-SALES-11667)~~



Drop ob

→ Alter table table\_name ADD (col1 datatype, col2 datatype);

→ Alter table tablename, MODIFY colname coltype;

→ alter table table-name change column old-name;  
To newname

→ Update table-name set —, where —; (more than 1 by comma)

→ Delete from table name where \_\_\_\_\_;



③

## Movie

✓ a) select mov\_title from movies m, directors d  
where d.dir\_id = m.dir\_id AND  
dis\_name = 'BUT';

my  
X b) Select m.movie\_title from movies m, <sup>mov</sup>cast c  
where m.mov\_id = c.mov\_id ~~AND~~ <sup>and Act ID in</sup> ~~Group~~  
BY c.act\_id having count(\*) > 1;

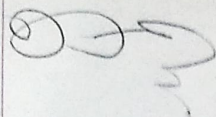
their  
✓ c) Select m.movie\_title from movies m, mov\_cast c,  
where m.mov\_id = c.mov\_id AND act\_id IN (  
select act\_id from mov\_cast ~~where~~ Group by  
act\_id having count(<sup>mov\_id</sup> ~~act\_id~~) > 1) ~~Group by~~  
~~mov\_title having count(\*) > 1;~~

X c) Select Actname, Act\_id, ~~mov~~title from  
Actor A JOIN MOVIEcast C ON A.act\_id =  
C.act\_id JOIN MOVIES M ON M.mov\_id =  
C.mov\_id where MOV\_year NOT BETWEEN  
2000 AND 2015;

Reimage?  
d) Select mov\_title, max(rev\_stars) from movies,  
rating where movies.mov\_id = rating.mov\_id group by  
mov\_title having max(rev\_stars) > 0 order by  
mov\_title;

✓ e) ~~Select~~ Update rating set Rev\_stars = 5  
where mov\_id in (select mov\_id from movies  
where dir\_id in (select dir\_id from  
director where dis\_name = 'xyz'));





- c) Select a. act name from (after a join move cost c on a. act id = c. act id) join moves m on m. mov id = c. mov id where mov year < 2000 and exists (select a. act name from (after a join move cost c on a. act id = c. act id) join moves m on m. mov id = c. mov id where mov year > 2015);

### (A) College

- a) Select S.\*, SS.SEM, SS.SEC from Student S, semsec SS, class C where S.USN = C.USN and SS.SSID = C.SSID and sem = 4 and sec = 'C';
- b) Select SS.sem, SS.sec, S.gender, count(S.gender) as count from Student S, semsec SS, class C where S.USN = C.USN and SS.SSID = C.SSID Group By SS.sem, SS.sec, S.gender.
- c) Create view STDMARKS1 AS select S.USN, S.name, SB.Subcode, SB.TITLE, I.TEST1 from Student S, Subject SB, I Marks I where S.USN = I.USN and SB.Subcode = I.Subcode and S.USN = '11X1234';
- d) Update I Marks set finalIA = (test1 + test2 + test3 - least(test1, test2, test3)) / 2;
- e) Select S.\*, (CASE WHEN IA.FINALIA Between 17 AND 20 THEN 'Outstanding' WHEN IA.FINALIA BETWEEN



12 and 16 then 'Average' ELSE 'WEAK'  
END) AS CAT

from ~~Students~~ Tamaras TA, SENSE SS,  
~~is~~ where ~~SS.SID = TA.DSID~~, SS.SID =  
14.SSID, ~~SS.SUBCODE = TA.SUBCODE~~ AND  
SS.SEM = 8 AND SS.SEC IN ('A', 'B', 'C');

### ⑤ Company

Work on primary key? all 3.

✓ a) Select distinct P.PNO from project P, dept D,  
emp E where E.dno = P.dno AND E.SSN =  
D.MGRSSN AND E.NAME LIKE '%SCOTT'.  
UNION

Select distinct P.PNO from project P, work on w,  
emp E where P.PNO = W.PNO AND E.SSN = W.SSN  
AND E.NAME LIKE '%SCOTT';

as —

✓ b) Select E.Name, E.Salary \* 1.1, from employee E,  
project P, work on W where E.SSN = W.SSN AND  
P.PNO = W.PNO AND P.PNAME = 'LOT';

✓ c) Select sum(salary), max(salary), min(salary),  
avg(salary) from emp dept where  
emp.dno = dept.dno AND dept.dname = 'Accounts';

✓ d) Select e.name from employee e where  
not exists (select pno from project where  
dno = 5 and pno not in (select pno from  
work on where e.ssn = w.ssn))



e) select d.dno, count(\*) from employee e, dept d.  
where e.dno = d.dno AND e.salary > 60000  
AND d.dno in (select e1.dno from employee e1  
group by e1.dno having count(\*) > 5)  
group by d.dno;