CSE 2007 — DBMS

JOB FERNANDEZ
19BCD7154

I.

① List all branches and their assets

```
select branch_name, assets from branch;
```

② List all accounts of Brooklyn branch.

```
select account_number from account natural
join branch where branch_city = "Brooklyn";
```

③ List all loans with amount > 1000

```
select * from loan where amount > 1000;
```

④ List all accounts of Perryridge balance with balance < 1000

```
select account_number from account
where branch_name = "Perryridge" and
balance < 1000;
```

⑤ List numbers of accounts with balances between 700 and 900

```
select count (account_number) from account
where balance > 700 and balance < 900;
```

⑥ Change the assets of Perryridge branch to 340000000

```
update branch set assets = 340000000
where branch_name = "Perryridge";
```

19BCD7154
Job

⑦ Transfer the accounts and loans of Perryridge branch to Downtown branch (subquery)

update amount, loan set branch_name = "Downtown" where = "Perryridge";

⑧ Transfer Rs. 100 from account A-101 to A215

update account set_balance = case
when account_number = "A-101" then balance = 100
when account_number = "A-215" then balance + 100
else balance
end;

⑨ Delete the branch Perryridge

delete from account, branch, loan where branch_name = "Perryridge";

⑩ Display all the loans with amount < 1000

select * from loans where amount < 1000

⑪ Delete all accounts and loans of Downtown branch

delete from depositor where account_number (select ~~loan_number~~ account_number from account where branch_name = "Downtown");
delete from account where branch_name = "Downtown";

delete from borrower where loan_number in [Job 19BCD7154
(select loan_number from loan where branch_name
="Downtown");

Job 19BCD7154

(12) Add a column phoneNo to the customer table

alter table customer phoneNo number(10);

(13) Change the size of branch_city to varchar(20)

alter table branch alter column branch_city
type varchar(20);

(14) Drop the column phoneNo from customer table

alter table customer drop phoneNo;

(15) List names of customers whose starting letter is 'S'

select customer_name from customer where
customer_name like ('S%');

(16) Drop the primary key constraint

alter table account drop constraint account_number;

11. Write queries for the following functions using Date function.

Job
19BCD7154

① Return the current date and time of the system in which the database is configured

```
select current_date ;
```

---

② Extract the day, month and year of the date 25-sep-1990

```
select dayname('1990-09-25'), monthname
('1990-09-25'), year ('1990-09-25')
```

---

③ Find the last date of this month

```
select last_day (SYSDATE);
```

---

④ Find the next Sunday date from the given date 11-Jul-2007

```
select current_date ( ) + interval 6 -
    weekday (current_date ())
```

---

⑤ Retrive the current system date and time which includes fractions of seconds and time zone

```
select SYSDATE();
```

Job 19BCD7154

⑥ Truncate the present date in case of month and year

```
select current_timestamp();
```

⑦ Display the current time stamp

```
select current_timestamp();
```

---

⑧ Convert the given character '01-Jan-2015' into the date

```
select str_to_date("1-January-2015",
                   "%d %M %Y");
```

---

⑨ Display the month in character of the given date 29-03-2010

```
select date_format("29-03-2010", "%M");
```

⑩ Display the day in character of the given day 01-02-2006

```
select date_format("2006-02-01", "%D");
```

---

⑪ Display the current year in character

```
select date_format(SYSDATE(), "%Y");
```

(12) Display the months between the present date and 17-Sep-2016

```
select datediff(month, SYSDATE(), "2016-09-17")
```

(13) Find the last day of this year 2015.

```
select last_day( date_add( now(),
interval 12 - month() month );
```

(14) Find how many days are left in the current month

```
select date_diff (last_day (sysdate()),
current_date());
```

(15) Find the first day of the month

```
select dateadd (month, datediff (month, 0,
@mydate), 0)
```

(16) How will you display the current user

```
select user();
```

Job 19BCD7154

III Consider the following relation Customer_Transaction

Relation: Table: Book_title

Job
19BCD7154

Title_id	Title	Type	Price	Published Year
BU1234	"The busy executives database guide"	Business	700	2001
BU4567	"Cooking with the computers"	Business	500	2000
M33458	"The Microwave"	Mod-cook	120	1998
PS1267	"Is Anger the Enemy"	Psychology	300	1898
BU3490	"Straight talk about computers"	Business	1000	2007

Write the queries from the following questions using multiple row function:

① List the minimum price from Book_title Table

```
select Price from Book_title where
Price = (select min(price) from Book_title);
```

② List the maximum price from Book_title table

```
select Price from Book_title where
Price = (select max(price) from Book_title);
```

Job
19BCD7154

③ List the total number of books

```
select count(*) as Title from Book_title;
```

④ List the total number of books whose price is more than 200

```
select Title, Price from Book_title where
Price > 200 order by Price desc, title;
```

⑤ List the minimum and sum of all book price from Book_title Table

```
select Price, Title, min(Price) as Minimum
Price and sum(Price) as Sum from
Book_title;
```

⑥ List the total number of books under the type Business

```
select count(*) from Book_title where
Type = "Business";
```

⑦ List the average price from Book_title Table

```
select Price, Title from Book_title where
Price = (select avg(Price) from Book_title);
```

⑧ List the title_id of the book that was published before 2000

Job 19BCD7154

```
select Title_id from Book_title where
Published_year < 2000;
```

IV Write a query to create table Payroll_details with Worker_id, name, consultancy_name and Hiredate, Salary as fields and insert the following values listed in below table (All three types of insertion)

Relation : Payroll_details

Worker_id	Name	consultancy_name	Hiredate	Salary
12f209	King	IBM	20-Jun-2012	5000
13e320	Nancy	Cognizant	13-Jan-2013	5060
15f345	Joe	Accenture	15-Sep-2015	7000
14f567	Robert	TCS	25-Sep-2014	8900
13f456	Anne	Infosys	17-Jul-2013	8000

Write the queries for the following statements.

① Increase the number of digits for Salary to 7.

```
alter table Payroll_details alter column
Salary type number(7);
```

Job 19BCD7154

② Add a new column DOB in the payroll_details table:

alter table Payroll_details add DOB varchar (10);

---

③ Update the values in DOB column.

insert into Payroll_details (DOB) values
("20-05-2011", "10-05-2003", "5-05-2001"
"10-11-2000", "25-05-2010", "30-05-2001");

---

④ Salary is incremented by Rs. 1000 per unit Display the incremented value.

update Payroll_details SET Salary=Salary+1000;

---

⑤ Combine NAME and CONSULTANCY_NAME and display it as EMP_NAME_CONSULTANCY

select concat (NAME, CONSULTANCY_NAME) as EMP_NAME_CONSULTANCY from Payroll_details;

---

⑥ The salary of King is raised to 10000.

update Payroll_details Salary = 10000
where name = "King";

---

⑦ Remove the content permanently from database

delete from Payroll_details;

(8) Remove the table permanently from database

drop table Payroll_details;

Job
19BCD7154

Consider the following relations

Relation 1: Dept.

DEPTNO	DNAME	LOC
10	Accounting	New York
20	Research	Dallas
30	Sales	Chicago
40	Operations	Boston

Relations 2: SalGrade

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Relation 3 : EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPT NO
7839	King	President		17-Nov-81	5000		10
7698	Blake	Manger	7839	01-May-81	2850		30
7782	Clark	Manager	7839	09-Jun-81	2450		10
7566	Jones	Manager	7839	02-Apr-81	2975		20
7788	Scott	Analyst	7566	19-Apr-87	3000		20
7902	Ford	Analyst	7566	03-Dec-81	3000		20
7369	Smith	clerk	7902	17-Dec-80	800		20
7499	Allen	Salesman	7698	20-Feb-81	1600	300	30
7521	Ward	Salesman	7698	22-Feb-81	1250	500	30
7654	Martin	Salesman	7698	28-Feb-81	1250	1400	30
7844	Turner	Salesman	7698	08-Sep-81	1500	0	30
7876	Adams	Clerk	7788	23-May-87	1100		20
7900	James	Clerk	7698	03-Dec-81	950		30
7934	Miller	Clerk	7782	23-Jan-82	1300		10

(BCDTIST JOB

JOB
19BCD7154

Answer the following using subquery

① Write a query to display the employee name and hiredate for all employees in the same department as Blake. Exclude Blake

```
select ENAME, HIREDATE from EMP where
DEPT_NO. IN (select DEPT_NO from
EMP where ENAME= "Blake") and ENAME
!= "Blake";
```

---

② Create a query to display the employee number and name for all employees who earn more than average salary. Sort the results in descending order of salary

```
select EMPNO, ENAME from EMP where
SAL > (select AVG(SAL) from EMP) order
by SAL desc;
```

---

③ Write a query the employee number and name for all employees who work in a department with any employee whose name contains a T

```
select EMPNO, ENAME from EMP where DEPT_NO
in (select DEPT_NO from EMP where ENAME
like "%T%");
```

JOB 19BCDT115

④ Display the employee name, department number and job title for all employees whose department location is Dallas

```
select ENAME, DEPT_NO , JOB from
EMP where DEPT_NO in (select DEPT_NO
from DEPT where LOC = "Dallas");
```

VI Consider the following relation

Relation : Pop

NAME	CONTINENT	AREA	POPULATION	GDP
Afghanistan	Asia	652230	25500100	20343000000
Albania	Europe	28748	2831741	12960000000
Algeria	Africa	2381741	37100000	188681000000
Andorra	Europe	468	78115	3712000000
Angola	Africa	1246700	20609294	100990000000

Answer the following using subquery

① List each country name where population is larger than 'Russia'.

```
select NAME from Pop where POPULATION
> (select POPULATION from Pop where
NAME = "Russia");
```

② Show the countries in Europe with a per capita GDP greater than 'United Kingdom'.

```
select NAME from Pop where CONTINENT =
"Europe" and GDP/Population > (select
```

GDP / POPULATION from Pop where NAME = 'United Kingdom');

JOB 19BCD7154

---

③ List the name and continent of countries in the continents containing 'Belize', 'Belgium'.

select NAME, CONTINENT from Pop where CONTINENT in (select CONTINENT from Pop where NAME = "Belize" or NAME = "Belgium");

---

④ Which country has a population that is more than Canada, but less than Poland? Show the name and Population.

select NAME, POPULATION from Pop where POPULATION > (select POPULATION from Pop where NAME = "Canada") and POPULATION < (select POPULATION from Pop where NAME = "Poland");

---

⑤ Find the largest country (by area) in each continent, show the continent, the name and the area

select X.CONTINENT, X.NAME, X.AREA from Pop as X where X.AREA = (select max (Y.AREA) from Pop as Y where X.CONTINENT = Y.CONTINENT);

JOB 19 BCD

⑥ Find the continents where all countries have a population <= 25000000. Then find the names of the countries associated with these continents.
Show name, continent & population.

```
select X.NAME, X.CONTINENT, X.POPULATION
from Pop as X where 25000000 >= all
(select Y.POPULATION from Pop as Y where
X.CONTINENT = Y.CONTINENT);
```

VII. Consider the relations

Employee (employee-name, street, city)
Works (employee-name, company-name, salary)
Company (company-name, city)
Manages (employee-name, manager-name)

Answer the following Joins

① Find the names, street address and cities of residence for all employees who work for 'First Bank Corporation' and earn more than $10,000

```
select Employee.employee_name,
Employee.street, Employee.city from
Employee left join Works on Employee
Employee.employee_name = Works.employee_name
where Works.company_name
```

= 'First Bank Corporation and Works. Salary >
10000;

JOB
19BCD7154

---

② Find the names of all employees in the database who live in the same cities as the companies for which they work.

```
select Employee.employee_name from
Employee left join Company on
Employee.city = Company.city;
```

---

③ Find the names of all employees in the database who live in the same cities and on the same streets as do their managers

```
select a.employee_name from Employee as
a left join Manager
a.employee_name = Manger.employee_name
left join Employee as b on a.employee_name
= b.employee_name where a.city =
b.city and b.employee_name =
Manager.employee_name and a.street =
b.street;
```

④ Find the names of all employees in the database who do not work for the 'First Bank Corporation'. Assume all the people work exactly for one company.

JOB
19BCD7154

```
select Employee.employee_name from Employee
left join Works on Employee.employee_name
= Works.employee_name where
Works.company_name != 'First Bank Corporation';
```

---

⑤ Find the names of all employees in the database who earn more than every employee of 'Small Bank Corporation'. Assume that all people work for exactly one company

```
select a.employee_name from Employee as a
left join Employee as b on a.employee_name
= b.employee_name left join Works on
Employee.employee_name = Works.employee_name
where a.salary > b.salary
and b.company_name = "Small Bank Corporation"
and a.company_name != b.company_name;
```

---

⑥ Assume that the companies may be located in several cities. Find all companies located in every city in which 'Small Bank Corporation' is located.

select a.company_name from company as a left join company as b on a.company_name = b.company_name where a.city = b.city and b.company_name = "Small Bank Corporation";

VIII Normalise the below property

Client no	CName	Property No	Paddress	Rentstart	Rentfinish	Rent	Owner No	ONa
CR76	John kay	PG4	6 Lawrence St. Glassgow	1-jul-03	31-Aug-04	350	CO40	Tina M
		PG16	5, Novar Dr Glasgow	1-sep-04	1-sep-05	450	CO93	Tony Sh
CR56	Aline Stewart	PG4	6 Lawrence St. Glassgow	1-sep-02	10-jun-03	350	CO40	Tina Mur
		PG36	2 Manor Rd, Glassgow	10-oct-03	1-dec-04	375	CO93	Tony Sha
		PG16	5 Novar dr, Glassgow	1-nov-05	10-aug-06	450	CO93	Tony Sha

JOB
19BCD7154

A → Client_No

B → CName

C → Property_No

D → Paddress

E → Rentstart

F → Rentfinish

G → Rent

H → Owner_No

I → Oname

F.D:

A → B

C → D

H → I

EF → G

$ABCDEFGHI^+ = \{A, B, C, D, E, F, G, H, I\}$

$ACEFGHI^+ = \{A, C, E, F, G, H, I, B, D\}$

$ACEFH^+ = \{A, C, E, F, H, B, G, D, I\}$

No proper subsets

$R_1(A, B)$

$A^+ = (A, B)$

$B^+ = (B)$

FD: A → B

A is

BCNF

$R_2(C, D)$

$C^+ = (C, D)$

$D^+ = D$

FD: C → D

C is

BCNF

$R_3(E, F, G)$

$E^+ = E$

$F^+ = F$

$G^+ = G$

FD: {} → null

BCNF

$R_4(H, I)$

$H^+ = \{H, I\}$

$I^+ = I$

FD: H → I

H is a super key here

BCNF

JOB
19BCD7154

ClientNO	CName
CR76	John Key
CR56	Aline Stewart

Property_No	Paddress
PG14	6 Lawrence St., Glassgow
PG16	5, Novar Dr, Glasgow
PG36	2 Manor Rd, Glasgow

Rentstart	Rentfinish	Rent
1-jul-03	31-aug-04	350
1-sep-04	1-sep-05	450
1-sep-02	10-jun-03	350
10-oct-03	1-dec-04	375
1-nov-05	10-aug-06	450

Owner_NO	OName
CO40	Tina Murphy
CO93	Tony Shaw

JOB 19BCD7154

(i) Normalise the following employee table

Relation: Employee

EmpNo	Surname	FirstName	DOB	QualCode	QualDesc	JobCode	JobDesc	CurrentDept	CurrentSalary
1234	Bloggs	Joe	23/6/53	BSCENG	BSc English	JPROG	Jnr Prog	Engg	23,500
				MSCPHY	MSc Physics	PROG	Programmer		
				PHDCS	PhD Computer Science	ANALP	Anal/Prog		
						PM	Project Manager		
1332	Jones	Mary	22/6/55	MSCPHY	MSc Physics	JPROG	Jnr Prog	R&D	32,450
				PHDCS	PhD Computer Science	ANALP	Anal/Prog		
4321	Smith	Tom	21/3/66	BSCF	BSc. French	JPROG	Jnr Prog	R&D	17,500

F.D.1 : Emp_No_Surname, FirstName, DOB, Job
Current_Dept, Current_Salary

19 BCD 7154

F.D.2 : Emp-No, Job_Code → Surname, FirstName,
DOB, Qual_Code, Qual_Desc, Job_Desc,
Current_Dept, current_Salary

FD3 : Empno, Job_Desc → Surname, First_Name
DOB, Qual_Code, Qual_Desc, Job_Code,
Current_Dept, Current_Salary

FD 4: Job_Code → Job_Desc

FD 5: Qual_Code → Qual_Desc

R1:

Emp_No	Job_Code	Qual_Code	Qual_Desc
1234	JRRCG	BSCENG	BSc English
1234	PROG	MSCPHY	MSc Physics
1234	ANALP	PHDCS	PhD Computer Sc.
1234	PM	MSCPHY	MSc Physics
1234	JRROG	PHDCS	PhD Computer Science
1332	ANALP	PHDCS	PhD Computer Sci
1332	JPROG	BSCF	BSc French.
4321			

R2

Emp_No	Surname	FirstName	DOB	Current_Dept	Current_Salary
1234	Bloggs	Joe	23-6-53	Engg	23500
1332	Jones	May	22-6-55	R & D	32456
4321	Smith	Tom	21-3-06	R&D	17500

R3:

Job_Code	Job_Desc
JPROG	Jnr Prog
PROG	Programmer
ANALP	Anal / Prog
PM	Project Manager

JOB
19BCD7154

3NF:

FDs : Qual_Code → Qual_Desc (Transitive Dependency)

(Seperating R1 into 2 tables)

R1:

Emp_no	Job_code	Qual_code
1234	JPROG	BSCENG
1234	PROG	MSCPHY
1234	ANALP	PHDCS
1234	PM	PHDCS
1332	JPROG	MSCPHY
1332	ANALP	PHDCS
4321	JPROG	BSCF

R2:

Qual_Code	Qual_Desc
BSCENG	BSc English
MSCPHY	Msc Physics
PHDCS	PhD Computer Science.
BSCF	BSc French.

∴ doesn't satisfy BCNF (max 3NF)

X Normalize it to give an efficient design

Job
19BCD7154

Rollno	Name	City	Subcode	Subname	Grade	Deptno	Deptname
101	XYZ	Coimbatore	12E502	DBMS	A	D1	EEE
101	XYZ	Coimbatore	12E506	OS	B	D1	EEE
102	WUV	Salem	12E503	DBMS	A	D1	EEE
102	WUV	Salem	12E506	OS	B	D1	EEE

- table is already in 1 NF
- 2NF (no PD)

FD1 : rollno, subcode → name, city, deptno, deptname subname, grade [PK]

FD2 : rollno → name, city, deptno, deptname [PD]

FD3 : rollno, subname → name, city, deptno, deptname, subcode, grade [CK]

FD4 : subcode → subname [PD]

R1

Rollno	Name	City	Deptno	Deptname
101	XYZ	Coimbatore	D1	EEE
102	WUV	Salem	D1	EEE

Job
19BCD7154

R2.

Rollno	Subcode	grade
101	12E503	A
101	12E506	B
102	12E503	A
102	12E506	B

R3:

Subcode	Subname
12E503	DBMS
12E506	OS

3NF (no TD)

FDS: Deptno → Deptname (transitive dependency)

(seperating R_1 into two tables)

R_1,

Rollno	Name	City	Deptno
101	XYZ	Coimbatore	D1
102	WVV	Salem	D1

$R1_2$

Deptno	Deptname
D1	EEE

R_2 : Same

Rollno	Subcode	Grade
101	12E503	A
101	12E506	B
102	12E503	A
102	12E506	B

R_3 (same)

Job
19BCD7154

Subcode	Subname
12E503	DBMS
12E506	OS

∴ table not normalized yet
BCNF possible – all determinants are super keys.

4NF:

R_2 shows multivalued dependency – splitting R_2

R_{21}

Rollno	Subcode
101	12E503
101	12E506
102	12E503
102	12E506

R_{22}

Rollno	Grade
101	A
101	B
102	A
102	B

∴ removing tables (R_{11}, R_{12}, R_3 stay the same)
∴ table normalized

#X] Consider the following relation & normalize the Office_details table.

Name	Project	Task	Office	Floor	Phone
Bill	100X	T1	400	4	1400
Bill	100X	T2	400	4	1400
Bill	200Y	T1	400	4	1400
Bill	200Y	T2	400	4	1400
Sue	100X	T33	442	4	1442
Sue	200Y	T33	442	4	1442
Sue	300Z	T33	442	4	1442
Ed	100X	T2	588	5	1588

JOB
19BCD7154

A → Name
B → Project
C → Task
D → Office
E → Floor
F → Phone

F.D.

A → DEF

$ABCDEF^+ = \{A, B, C, D, E, F\}$ → super key

$ABC^+ = \{A, B, C, D, E, F\}$ → C.K

No proper subsets that are super keys

D.A → {A, B, C}

There are no prime attributes that appears at the RHS of any FDs. So we take A, B, C is the only candidate key

R_1 (A, D, E, F)

$A^+ = \{A, D, E, F\}$ → S.K

$D^+ = \{D\}$

$E^+ = \{E\}$

$F^+ = \{F\}$

JOB
19BCD7154

F.D.

A → DEF

S.K in LHS:
of all FDs

BCNF

R_1 (A B C)

$A^+ = \{A\}$

$B^+ = \{B\}$

$C^+ = \{C\}$

FD = { }

⇒ B,C,N,F

JOB 19BCD7154

Name	Office	Floor	Phone
Bill	400	4	1400
Sue	442	4	1442
Ed	588	5	1588

BCNF

Name	Project	Task.
Bill	100X	T1
Bill	100X	T2
Bill	200Y	T1
Bill	200Y	T2
Sue	100X	T33
Sue	200Y	T3
Sue	300Z	T3
Ed	100X	T2

12. Consider the relation R(A B C D E F G) with the following set of functional dependencies:

$AD \rightarrow F$, $AE \rightarrow G$, $DF \rightarrow BC$, $E \rightarrow C$, $G \rightarrow E$

① List all the candidate keys (not super keys)

② Consider the decomposition of relation R into 4 relations: R1(ADF), R2(CE), R3(EG) and R4(ABDG) What is the highest normal form of this decomposition? Is this decomposition dependency preserving? Is it loseless? Justify your answer.

$AD \rightarrow F$

$AE \rightarrow G$

$DF \rightarrow BC$

$E \rightarrow C$

$G \rightarrow E$

① Here the candidate keys are ADE and ADG

. A & D must be there in any candidate key as no other attribute determines them

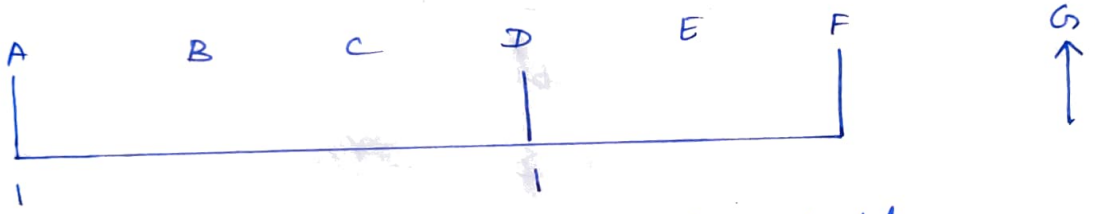
$(AD)^*$. ADFBC

. So we still need to cover E & G which are not determined by A & D

JOB
19BCD7154

$(ADE)^* = ABCDEFG$

$(ABG)^* = ABCDEFG$.



② From the given decomposition table.

(A D F) (C E) (E G) (A B D G)

BCNF

BCNF

BCNF

partial dependency on candidate key ADG.
Hence, it's not in 2NF

∴ the highest form of deconposition in 1NF

The decomposition is not dependency preservating as following:

functional dependies from initial set aren't preserved

JOB
19BCD7154

$AE \longrightarrow G$

$DF \longrightarrow BC$

The decomposition is not dependency preserving as following functional dependencies from the initial set are not preserved.

$AE \longrightarrow G$

$DF \longrightarrow BC$

The decomposition is loseless as the following order of joining will bring back the original table

$$(A\ D\ F) \bowtie (A\ B\ D\ G) = (A\ B\ D\ F\ G)$$

Intersection AD determines the left table implies join

$$(C\ E) \bowtie (E\ G) = (C\ E\ G)$$

Intersection E determines the left table implies loseless join

$$(A\ B\ D\ F\ G) \bowtie (C\ E\ G) = (A\ B\ C\ D\ E\ F\ G)$$

Intersection G determines the right table, implies loseless join

Here, the given decomposition is loseless

XIII Following is the set of functional dependencies on the relational schema (P, Q, R)

JOB
19BCD1154

$P \rightarrow QR$, $PQ \rightarrow R$, $Q \rightarrow R$, $P \rightarrow Q$.

Find its minimal cover

$$f = \{ P \rightarrow QR, \quad PQ \rightarrow R, \quad Q \rightarrow R, \quad P \rightarrow Q \}$$

(i) split rule

$$\{ P \rightarrow Q, P \rightarrow R, PQ \rightarrow R, Q \rightarrow R, P \rightarrow Q \}$$

(ii) remove extranious attribute

$$\{ P \rightarrow Q \;, \cancel{P \rightarrow R} \;, P \rightarrow R, Q \rightarrow R, \cancel{P \rightarrow Q} \}$$

(iii) remove redundent or duplicate

$\{ P^+ = PQR$ so $P \rightarrow Q$ & $Q \rightarrow R$

$\{ P \rightarrow Q, Q \rightarrow R \}$

$P^+ \rightarrow PQR$

$Q^+ \rightarrow Q$

11/11

Consider the following sets of functional dependencies over a relation R(A, B, C)

JOB
19BCD7154

$F1 = \{ A \to B, B \to C \}$

$F2 = \{ A \to B, A \to C \}$

$F3 = \{ A \to B, AB \to C \}$

Which of these sets are equivalent?

$F1 = \{ A \to B, B \to C \}$

$F2 = \{ A \to B, A \to C \}$

$F3 = \{ A \to B, AB \to C \}$

F1 & F2

$F1 \Rightarrow A^+ = \{ A, B, C \}$

$A^+ = \{ B \}$

$F2 \Rightarrow A^+ = \{ A, B, C \}$

In F1, B derives C but in F2

B derives only B

Hence $F1 \neq F2$

F2 & F3

F2

$A^+ = \{ A, B, C \}$

F3

$A^+ = \{ A, B, C \}$

$AB^+ = \{ A, B, C \}$

$\Rightarrow$ F2 $\subseteq$ F3 & F3 $\subseteq$ F2

Hence F2 = F3

JOB
19BCD7154

F1 & F3

F1 $\Rightarrow$ $A^+ = \{A, B, C\}$

$B^+ = \{B\}$

F3 $\Rightarrow$ $A^+ = \{A, B, C\}$

$AB^+ = \{A, B, C\}$

B derives C in F1, but this is not satisfied by F3

$\Rightarrow$ F1 $\supseteq$ F3 & F1 $\not\subseteq$ F3

$\therefore$ F1 $\not\equiv$ F3

Hence the only set that are equivalent are F2 & F3

XV) Find the minimal cover of the subset of functional dependencies given

$\{A \rightarrow C, AB \rightarrow C, C \rightarrow DE, CD \rightarrow E\}$

$A \rightarrow C$

$AB \rightarrow C$

$C \rightarrow DE$

$CD \rightarrow E$

(i) RHS, making a single attribute

$A \rightarrow C$
$AB \rightarrow C$
$C \rightarrow D$
$C \rightarrow E$
$CD \rightarrow E$

JOB
19BCD7154

(ii) removing extreneous attribute

$A \rightarrow C$
$A \rightarrow C$
$C \rightarrow D$
$C \rightarrow E$
$C \rightarrow E$

(iii) remove reductant F.D

$A \rightarrow C$
$C \rightarrow D$
$C \rightarrow E$

XVI Let $R = ABCDE$. Let the functional dependencies be $A \rightarrow C$, $B \rightarrow C$, $C \rightarrow D$, $DE \rightarrow C$, $CE \rightarrow A$ Find whether the following decomposition is a lossless or lossy decomposition

$R1 = AD$, $R2 = AB$, $R3 = BE$, $R4 = CDE$ and $R5 = AE$

Soln:

	A	B	C	D	E
R1(AD)	a_1	b_{12}	b_{13}	a_4	b_{15}
R2(AB)	a_1	b_{23}	b_{23}	b_{24}	b_{25}
R3(BE)	b_{31}	b_{33}	b_{33}	b_{34}	a_5
R4(CDE)	b_{41}	a_3	a_3	a_4	a_5
R5(AE)	a_1	b_{53}	b_{53}	b_{54}	a_5

JOB 19BCD7154

by F ∞ A → C 125 need to change rules of RHS

A	B	C	D	E
a_1	b_{12}	b_{13}	a_4	b_{15}
a_1	a_2	b_{13}	b_{24}	b_{25}
b_{31}	a_2	b_{33}	b_{34}	a_5
b_{41}	b_{42}	a_3	a_4	a_5
a_1	b_{52}	b_{13}	b_{54}	a_5

now by applying B → C

A	B	C	D	E
a_1	b_{12}	b_{13}	a_4	b_{15}
a_1	a_2	b_{13}	b_{24}	b_{25}
b_{31}	a_2	b_{13}	b_{34}	a_5
b_{41}	b_{42}	a_3	a_4	a_5
a_4	b_{52}	b_{13}	b_{54}	a_5

next use C → D equate a_4, b_{24} b_{34} and b_{54}

A	B	C	D	E
a_1	b_{12}	b_{13}	a_4	b_{15}
a_1	a_2	b_{13}	a_4	b_{25}
b_{31}	a_2	b_{13}	a_4	a_5
b_{41}	b_{42}	a_3	a_4	a_5
a_1	b_{52}	b_{13}	a_4	a_5

DE → C helps equate b_{13} with a_3

A	B	C	D	E
a_1	b_{12}	a_3	a_4	b_{15}
a_1	a_2	a_3	a_4	b_{25}
b_{31}	a_2	a_3	a_4	a_5
b_{41}	b_{42}	a_3	a_4	a_5
a_1	b_{52}	a_3	a_4	a_5

JOB
19BCD7154

Apply $CE \rightarrow A$, to equate b_{31}, b_{41}, a_1

A	B	C	D	E
a_1	b_{12}	a_3	a_4	b_{15}
a_1	a_2	a_3	a_4	b_{25}
a_1	a_2	a_3	a_4	a_5
a_1	b_{42}	a_3	a_4	a_5
a_1	b_{52}	a_3	a_4	a_5

Middle rows is all a's.
It has lossless join

XVII Consider the relation $R(A, B, C, D, E, G)$ with the following FDs

$$FD = \{ AB \to C,\ D \to EG,\ C \to A,\ BE \to C,\ BC \to D,\ CG \to BD,\ ACD \to B,\ CE \to AG \}$$

Find whether the following statements are true or false with justification

JOB
19BCD7154

a) the closure of BC is $\{A, D, E, G\}$

b) all attributes of R are in the closure of BC

c) the closure of AC is $\{A, C\}$

d) ABC is a super key

e) BC is the only candidate key of R

Answer:

a) $BC^+ = \{B, C, A, D, E, G\}$

False, The closure of BC^+ is not $\{A, BD, E, G\}$

---

b) True

$BC^+ = \{B, C, A, D, E, G\}$

BC is a super key since it derives all other attributes. Further it is a candidate key, since no proper subsets of this super key is a super key

JOB 19BCD7154

c) True

'A' alone cannot determine any attribute
'C' can only determine A & C
So AC together derive any other attribute
Hence $AC^+ = \{A, C\}$

---

d) True

BC derives D. D derives E & G

$\Rightarrow ABC^+ = \{A, B, C, D, E, G\}$

Since this closure contains all the other attributes, ABC is a superkey

---

e) False

$BC^+ = \{B, C, A, D, E, G\}$

This is a super key and has no proper subsets which is a super key.
Therefore BC is a candidate key

C appears in the RHS of $AB \rightarrow C$
Hence AB is a superkey
AB does not have a proper subset which is a superkey. Hence, it is a candidate key.

XVIII Consider the relation R(A, B, C, D, E, F) with the following FDs

$$FD = \{AB \to C,\ BC \to AD,\ D \to E,\ CF \to B\}$$

JOB
19BCD7154

Answer the following

① Does $AB \to D$ is valid?

② If valid, find the highest normal form of the relation R including $AB \to D$

Answer:

R(A B C D E F)

$$FD = \{AB \to C,\ BC \to AD,\ D \to E,\ CF \to B\}$$

① $AB \to D$ is valid

$(AB)^{\alpha} \longrightarrow ABCDE$

②

$(AF)^{+} \to X$

$(BF)^{+} \to X$

$(CF)^{+} \to B\ (\neq R)$

$(DF)^{+} \to F\ (\neq R)$

$(EF)^{+} \to X$

$(ABF)^{+} \to CDE = R$

$(ACF)^{+} \to BDE = R$

$(ADF)^{+} \to E \neq R$

$(AEF)^{+} \to X$

CK = ABF & ACF

∴ It is in 1NF

XIX Consider the following SQL query on the given set of schemas:

Student (sid, name, age, address)
Book (bid, title, author)
Checkout (sid, bid, date)

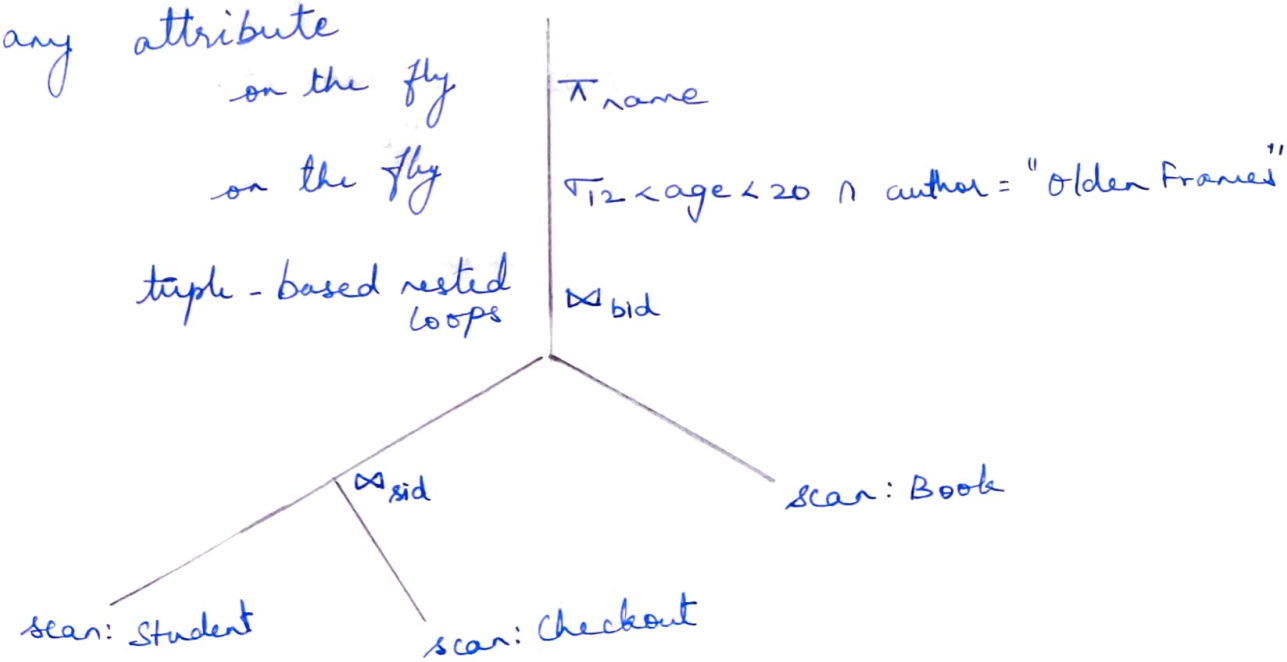
JOB
19BCD7154

```
SELECT S.name FROM Student S, Book B,
Checkout C WHERE S.sid = C.sid AND
B.author = "Olden Frames" AND S.age > 12 AND
S.age < 20
```

Show an expression tree for this query, assuming there are no indexes and data not stored on any attribute.

Answer:

Query for this, assuming there are no indexes and date isn't stored on any attribute



XX Consider the following SQL query on the schema
branch (branch_name, branch_city, assets)

select t.branch_name from branch t, branch s where t.assets > s.assets and s.branch_city = "Burnaby"

Write an efficient relational algebra expression that is equivalent to this query and justify your choice with an explanation

JOB
19BCD7154

Answer:

select t.branch_name

Π t.branch_name

where t.assets > s.assets &

branch_city = "Burnaby";

σ t.assets > s.assets AND s.branch_city = "Burnaby";

* Here, we have to rename branch as t & s to reduce ambiguity
* So, we can acheive these requisites by selecting from the left join of the branch t on branch s where t.branch_name = s.branch_name

Finally

$\Pi_{t.branch}$ ($\sigma_{(assets > s.assets}$ (P_t (branch) $\bowtie$ P_s (branch)) t.branch_name = s.branch_name AND s.branch_name = "Burnaby");

Define the schedule S for the given transactions. Check whether schedule S is view serializable or not.

T1	T2	T3	T4
	R(X)		
		W(X)	
		Commit	
W(X)			
Commit			
	W(Y)		
	R(Z)		
	Commit		
			R(X)
			R(Y)
			Commit

JOB
19BCD7154

Answer.

$R_2(x), W_3(x)$ $(T_2 \rightarrow T_3)$

$R_2(x), W_1(x)$ $(T_2 \rightarrow T_1)$

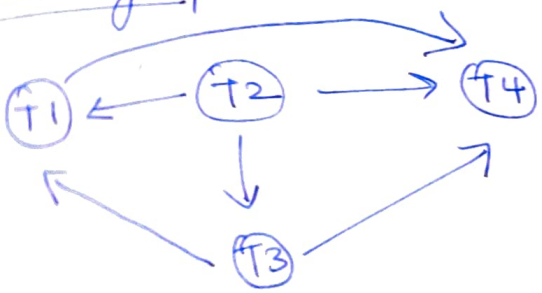
$W_3(x), W_1(x)$ $(T_3 \rightarrow T_1)$

$W_3(x), R_4(x)$ $(T_3 \rightarrow T_4)$

$W_1(x), R_4(x)$ $(T_1 \rightarrow T_4)$

$W_2(Y), R_4(Y)$ $(T_2 \rightarrow T_4)$

Precedence graph.



Clearly, there exists no cycle in precedence graph.

JOB
19BCD7154

∴ it is conflict serializable.

XXII. Consider the following schedule S involving two transactions T1 and T2. Only read and write operations have been shown. The read operation on data item P is denoted by Read(P) and write operation on data item P is denoted by Write(P). Is the schedule S recoverable if transaction T1 aborts at time instance 10. Justify your answer with a brief explanation.

Time Instance	T1	T2
1	Read (A)	
2	Write (A)	
3		Read (C)
4		Write (C)
5		Read (B)
6		Write (B)
7		Read (A)
8		Commit
9	Read (B)	
10	Abort	

If transaction fails it requires the effect of transaction to be undone.

Durability states that some transaction commits its change, it cannot be undone

JOB 19BCD7154

Recoverable schedule: A schedule exactly where for every set of transaction $T_i \rightarrow T_j$. T_i and T_j reads a item previously written by T_i, then the commit operation of T_i precedes the commit operation of T_j

The schedule S is non recoverable and cannot ensure transaction atomicity. This schedule is irrevocable by defination and also not atomic since it leaves the database in an inconsistent state

It's simply a dirty read, so non-recoverable.

XXIII For below given schedule draw the precedence graph and decide if the schedule is conflict serializable. If it is, give an equivalent serial schedule of transactions (eg) T2, T1, T3) If the schedule is not conflict serializable, explain (very briefly) why not label the edges of the precedence graphs with the data that causes the conflict. (eg) X, Y or Z)

S: $r_2(Y)\ w_2(Y)\ r_3(Y)\ r_1(X)\ w_1(X)\ w_3(Y)\ r_2(X)$
$r_1(Y)\ w_1(Y)$

JOB 19BCD7154

T1	T2	T3
	$r_2(Y)$	
	$w_2(Y)$	
		$r_3(Y)$
$r_1(X)$		
$w_1(X)$		
		$w_3(Y)$
	$r_2(X)$	
$r_1(Y)$		
$w_1(Y)$		

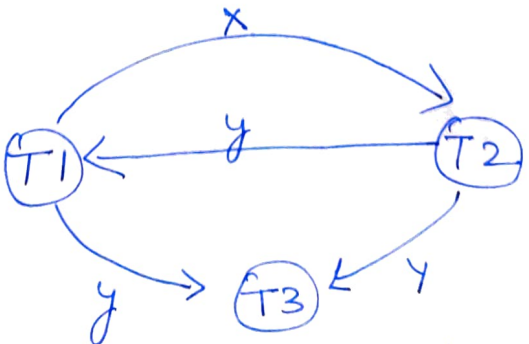
$T_1 \rightarrow T_2$: T_1 writes X before T_2 read X

$T_2 \rightarrow T_1$: T_2 writes Y before T_1 read Y

$T_3 \rightarrow T_1$: T_3 wrote Y before T_1 read Y

$T_2 \rightarrow T_3$: T_2 wrote Y before T_3 read Y

Graph:



There exists a schedule that is not conflict serializable

XXIV Assume an immediate database modification scheme. Consider the following log consisting transactions T1, T2 and T3:

1. (start, T1);
2. (Write, T1, P, 500, 600);
3. (Write T1, Q, 400, 500);
4. (commit, T1);
5. (start, T2);
6. (Write, T2, P, 600, 550);
7. (Write, T2, Q, 500, 450);
8. (commit T2)
9. (start T3)
10. (Write, T3, P, 550, 600);
11. (Write, T3, Q, 450, 500);
12. (commit, T3)

JOB
19BCD7154

If the schedule crashes just after step 7, then what will be the order of undo and redo actions performed during recovery process?

Solution:

If the schedule crashes after step 7, then the recovery operations are performed according to the order

REDO (T1), UNDO (T2)

as it's incurred because for the remaining algorithm, undo operations are done first and then redo operations.

XXV Consider the three transactions T1, T2, T3 and the schedules S1 and S2 given below

T1: r1(X); r1(Z); w1(X);
T2: r2(Z); r2(Y); w2(Z); w2(Y);
T3: r3(X); r3(Y); w3(Y)
S1: r1(X); r2(Z); r3(X); r3(Y); w1(X); w3(Y); r2(Y) w2(Z); w2(Y);

Draw the serializability (precedence) graphs for S1 & S2 and state whether each schedule is serializable or not.

JOB
19BCD7154

Answer:

T1	T2	T3
$r_1(X)$	$R_2(Z)$	$r_3(Y)$
$r_1(Z)$	$r_2(Y)$	$r_3(Y)$
	$W_2(Z)$	
$W_1(X)$	$W_2(Y)$	$W_3(Y)$

Schedule S1

T1	T2	T3
$r_1(X)$		
	$r_2(Z)$	
$r_1(Z)$		
		$r_3(X)$
		$r_3(Y)$
$W_1(X)$		
		$W_3(Y)$
	$r_2(Y)$	
	$W_2(Z)$	
	$W_2(Y)$	

Schedule S2

JOB
19BCD7154

T1	T2	T3
$r_1(X)$	$r_2(Z)$	$r_3(X)$
$r_1(Z)$	$r_2(Y)$	$r_3(Y)$
$w_1(X)$	$w_2(Z)$	
	$w_3(Y)$	
	$w_2(Y)$	

XXVI Consider the following schedule

- S1 : r1(X), w1(X), r2(X), r2(y), w2(X), c2, a1
- S2 : r1(X), r2(X), r3(X), w1(X), c2, a1
- S3 : r2(X), r1(X), w1(X), w2(X), a1
- S4 : r1(X), r1(y), w1(X), w1(y), r2(X), w2(X), c2, c1

(a) Is S1 recoverable? Justify.

(b) Is S2 cascadeless? Justify.

(c) Is S3 a strict schedule? Justify

(d) Is S4 irrecoverable? Justify.

Answer:

(a)

T1	T2
read(X)	
write(X)	
	read(X)
read(X)	
	write(X)
	commit;
abort;	

S1 is not recoverable.

In this the schedule adds a value uploaded by another transaction commit. Only after commit of transaction that uploaded the value, then the schedule is recoverable

Here, commit of T_2 comes first.
Hence, it is non-recoverable.

JOB
19BCD7154

(b)

T1	T2	T3
$r_1(X)$		
	$r_2(X)$	
		$r_3(X)$
$w_1(X)$		
	c_2	
a_1		

T2 accessed the data of T_1 before it was written. But T2 read x from T1 which was not uncompleted at the time. Since, there is a dirty read, the schedule isn't cascadeless

(c)

T1	T2
	$r_2(x)$
$r_1(X)$	
$w_1(X)$	$w_2(X)$
a_1	

This is not a strict schedule because T_1 can't use 'x' data of till T_2 commits
This condition has been violated and hence it isn't strict.

(d)

T1	T2
$r_1(x)$	
$r_1(y)$	
$w_1(x)$	
	$r_2(x)$
	$w_2(x)$
	c_2
c_1	

JOB
19BCD7154

This is irrecoverable. T2 accessed the value x after T2 updated it and T2 committed. Before T1 had committed. Hence it is irrecoverable

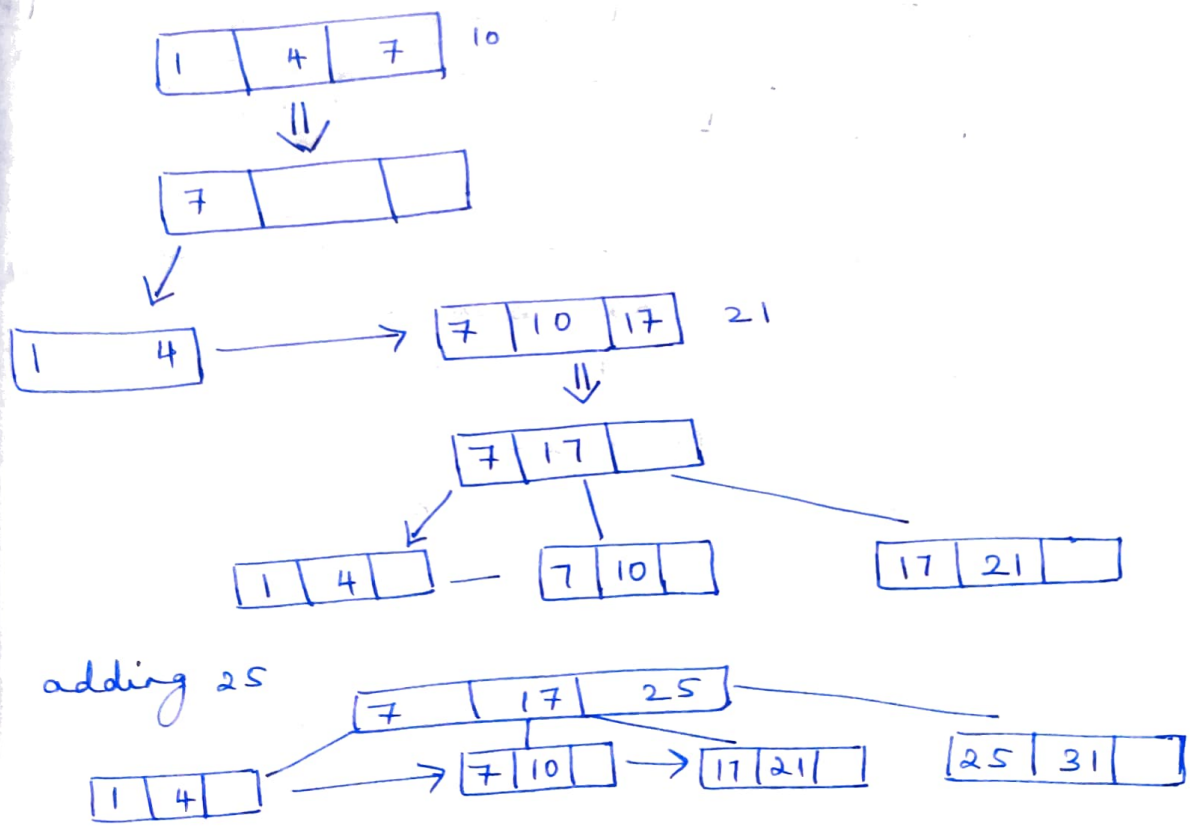
---

XXVII Construct a B+ tree for (1, 4, 7, 10, 17, 21, 31, 25, 19, 20, 28, 42) with n=4.

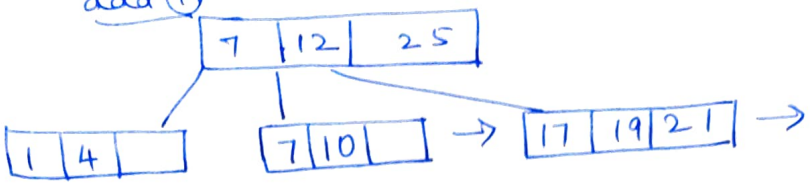
order $n = 4$

max. children = 4 max keys = $n-1 = 3$

min. children = $\frac{n}{2} = 2$ min keys = $\frac{n}{2} + 1 = 1$



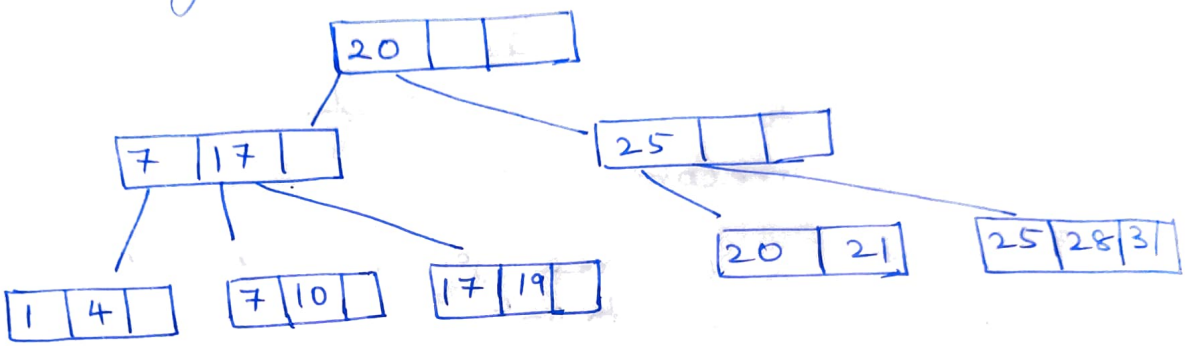
add (19)



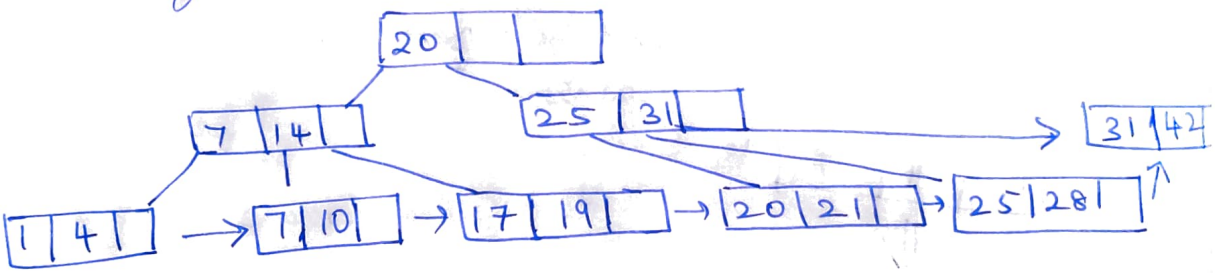
JOB
19BCD7154

In interal order, so data won't repeat

adding 20 & 28



adding 42



XXVIII. Suppose that we are using extendable hashing on a file that contains records with the following search-key values

2, 3, 5, 7, 11, 17, 19, 23, 29, 31

Show the construction of extendible hash structure by adding each given record for this file, if the hash function is $h(x) = x \bmod 8$ and buckets can hold three records

Answer:

(a) hash function is $h(x) = x \bmod 8$

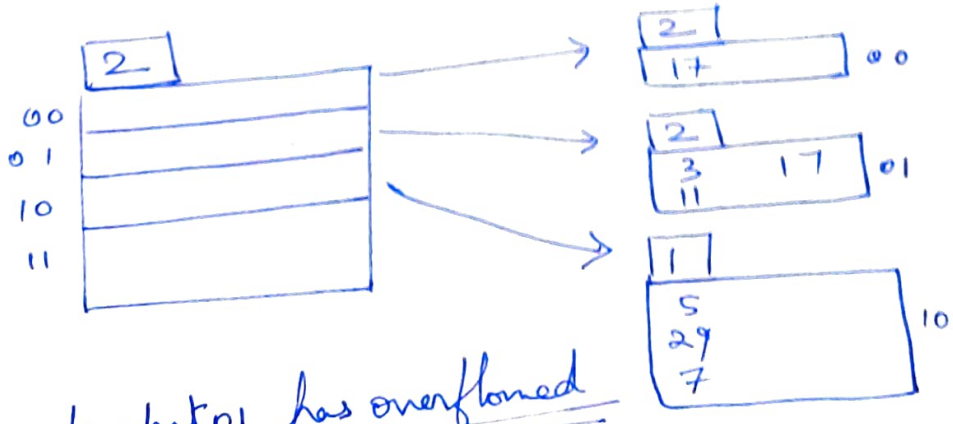
JOB
19BCD1154

Decimal	Binary
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1

Add 2 = 2 mod 8 = 010 → bucket 0
Add 3 = 3 mod 8 = 011 → bucket 0
Add 5 = 5 mod 8 = 101 → bucket 1
Add 7 = 7 mod 8 = 111 → bucket 1
Add 11 = 11 mod 8 = 011 → bucket 0
Add 17 = 17 mod 8 = 001 → bucket 00
Add 19 = 19 mod 8 = 011 → bucket 01
Add 23 = 23 mod 8 = 111 → bucket 11
Add 29 = 29 mod 8 = 101 → bucket 10
Add 31 = 31 mod 8 = 111 → bucket 11

Initial.

when 2 is added



bucket 01 has overflowed

XXIX Construct B tree of order 3 for the key
10, 20, 30, 40, 50, 60, 70, 80, 30, 35

Answer:

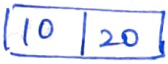
Keys = {10, 20, 30, 40, 50, 60, 70, 80, 30, 35}

$\wedge$ = 3

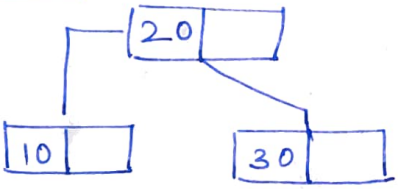
JOB
19BCD7154

1. Add 10

2. Add 20

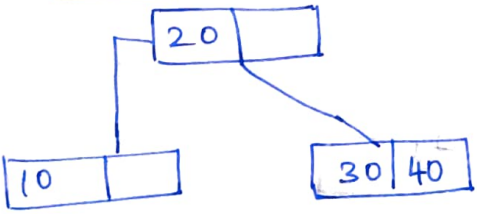


3. Add 30

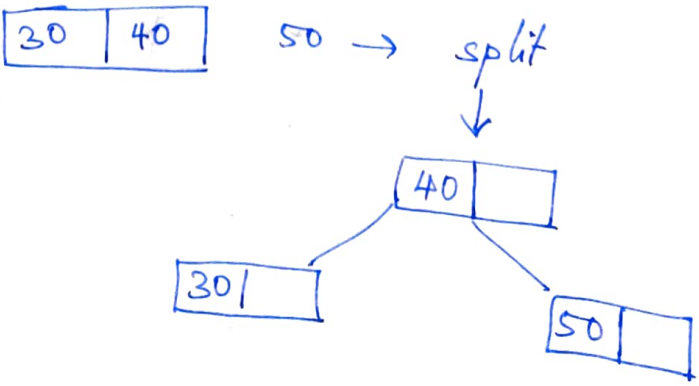


Right-biased.

4. Add 40

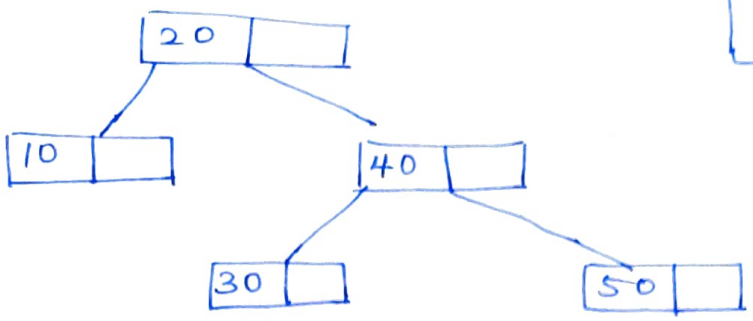


5. Add 50

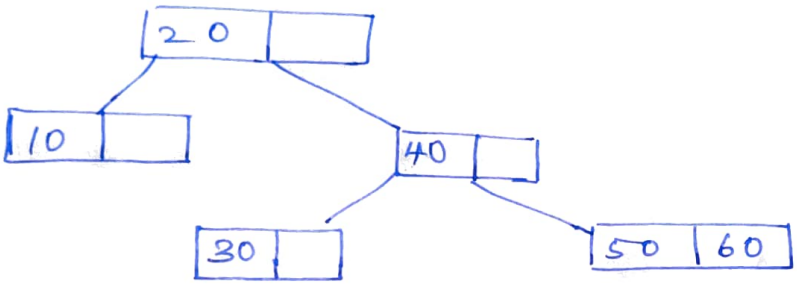


Hence,

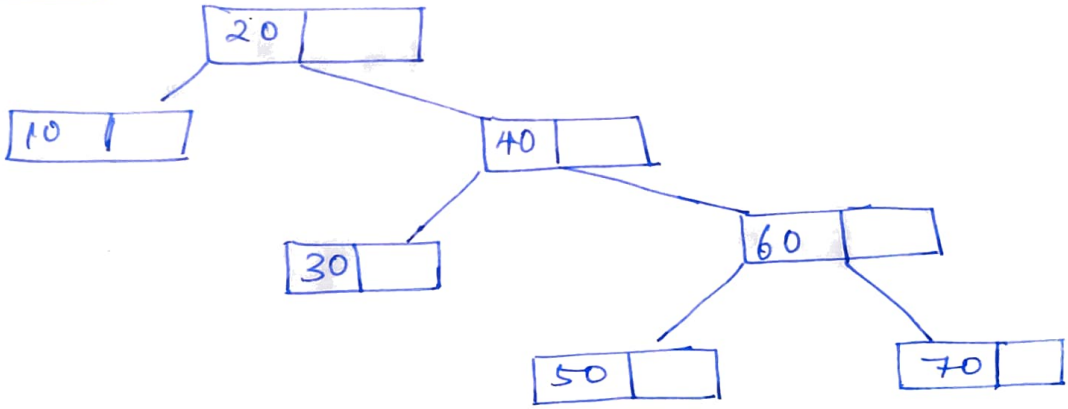
JOB
19BCD7154



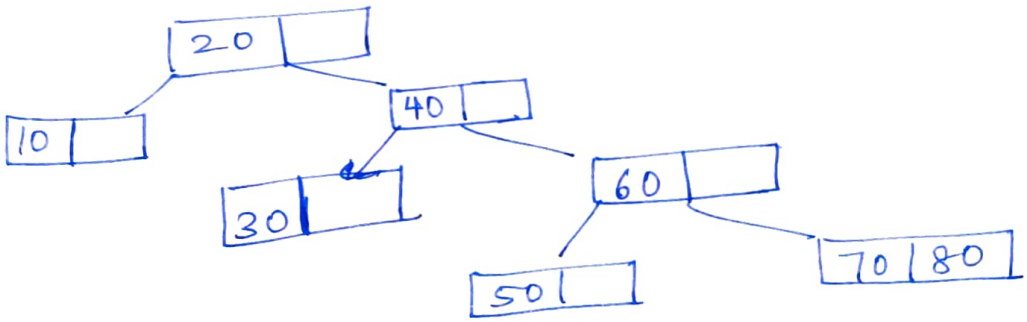
6. Add 60



7. Add 70



8. Add 80



9. Add 35

JOB
19BCD7154