



Insurance database - I (SPL (Cseittech))

- Create table person (driver_id varchar(15), name varchar(10), address varchar(20), primary key (driver_id));
- Create table car (regno varchar(15), model varchar(10), year int, primary key (regno));
- Create table Accident (report_number int, date int, location varchar(20), primary key (report_number),
foreign key (report_number) references accident);
- Create table N (driver_id varchar(10), Regno varchar(15), report_number int, damage_amount int, primary key (driver_id, regno), foreign key (driver_id) references person (driver_id), foreign key (regno) references car (regno), foreign key (report_no));
- Create table owns (driver_id int, Regno varchar(15), primary key (driver_id, Regno), foreign key (driver_id) references person (driver_id), foreign key (Regno) references car (regno));

→ (1) Enter 5 tuples (row) :-

→ insert into person values ('N01', 'Richard',
'Sonivas Nagar');

:

:

:

→ insert into car values ('KA081181', 'Lancer',
1957);

:

:

→ insert into account values (11, 2021/01/03, 'Southend',
'circle');

:

:

:

→ insert into purchased values ('N01', 'KA052250',
11, 25000);

→ \checkmark insert into owns account ('N01', 'KA052250');
11, 25000;

:

:

:



- (ii) update participated
set damage_amount = 2500 where reg_num =
[redacted]
- (iii) select count(*) from accident where year(acc_date) = 2008;
- (iv) select count(*) from participated where
reg_num in (select reg_num from car where
model = "Accord");

II

BANKING INSURANCE

~~Q~~

Branch (branchname (String, city, address)

Ans SQL : →

(i)

→ Create branch (branch_name varchar(10),
city varchar(10), primary
key (branch_name));

→

Create table bankaccnt (accno int,
branch_name varchar(25), balance int,
primary key (accno), foreign key (branch_name)
references branch (branch_name));

→ Create table bank_customer (customer_name varchar(25),
customer_street varchar(25), customer_city(15),
primary key (customer_name));

→ Create table depositor (customer_name varchar(25),
accno int, primary key (customer_name references
bank_customer (customer_name)),
branch_name);

→ Create table loan (loan_number int, branch_name
varchar(25), amount int, primary key (loan_number),
foreign key (branch_name) references branch
(branch_name));

Query 1 :- populating ↑

Query 3 :-

select distinct c.customer_name from bank_customer c,
bank_account b where exists (select d.customer_name
count (d.customer_name) from depositor d, bank
account ba where ba.accno = d.accno and
c.customer_name = d.customer_name and ba.branch
-name = 'SBI_ResidencyRoad' group by d.customer
name having count (d.customer_name) >= 2)

Query 5 :-

delete from bank_account where branch_name in
(select branch_name from branch where branch
city = 'Bombay');



→ Lab-3 : supplier database

(Create database supplier)

(i) → Create table supplier (S1D int primary key, Ename varchar(20), address varchar(20))
values

Insert into supplier ('1001', 'Acme Widg.', 'Bl.
Insert into supplier values('1002', 'Johni', 'Kota
Insert into supplier values('1003', 'Nimai', 'mumba
Insert into supplier values('1004', 'Reliance', 'Dell')

→ Create table parts (PID int primary key, Part
name varchar(30), @color varchar(30));

Insert into parts values (20001, 'Book', 'Red');
Insert into parts values(2002, 'Pen', 'Red');
Insert into parts values (2003, 'Pencil', 'Green')

→ Create table catalog (cost real, foreign key(S1D)
references supplier (S1D), foreign key(PID)
references parts (PID));

Insert into catalog (values (10, 1001, 2001)
Insert into catalog values (10, 1001, 2002);
Insert into catalog values (10, 10001, 1003);

Query

* Select supplier.name from supplier where
 supplier.supplier_id in (select catalog_id from
 catalog inner join parts on catalog.part_id
 = parts.part_id group by catalog.supplier_id
 having count(*) = (select count(part_id))
 from parts) ;

* Select supplier.name from supplier where
 supplier.supplier_id in (select catalog_id from
 catalog inner join parts on catalog.part_id
 = parts.part_id where catalog.part_id in (select
 part_id from parts where part.colour = 'Red')
 group by catalog.supplier_id having count(*) =
 select count(part_id) from parts where
 part.colour = 'Red') ;

Select parts.part_name from parts where parts
 part_id in (select catalog.part_id from catalog
 inner join parts on catalog.part_id = parts.part_id
 where catalog.supplier_id in (select supplier.supplier_id from
 supplier where supplier.name = 'Ane wedge'))



DBSS - LAB - 5

Ques 6

Create database Student Faculty

→ Create table student (snum int, sname varchar(30),
mname varchar(30), age int, primary key(snum))

→ Create table class (cname varchar(30), Time int, room
char(30), Fid int, primary key(cname),
foreign key(Fid) references faculty(Fid))

→ Create table faculty (Fid int, fname varchar,
dname int, primary key(Fid))

→ Create table enrolled (snum int, cname varchar
(30), primary key(snum, cname),
foreign key(Cname) references class(sname),
foreign key(snum) references student(snum))

Inserting values:-

Insert into student values (1, 'JON', 'IS', 'JPR', 18
(2, 'S', 'JE', 10), (3, 'JACOB', 'CN', 'SE', 20), (4, 'TOM',
'S', 'SE'), (5, 'RAHUL', 'C', 'JE', 20), (6, 'ELITA',
'S', 'SE', 21)).

Insert into class values ('12/11/13 10:15:16',
R127, 14), ('12/11/15 10:15:16', R128, 15),
(3, '12/11/15 10:15:20', R129, 15)

Ques:

- (1) Select distinct s.sname from student s, Faculty class c, enabled E where s.Snum = E.Snum and E.cname = C cname and C.FID = F.GID and F.fname = 'Harish' and s.fname = 'TV'
- (2) Select cname from class C where C. room = 125 or C cname in (select E cname from Enabled E)
 Group by cname having count (*) > 5;
- (3) Select distinct F.fname from faculty F where not exists (select C.room from class C minos
 Select C.room from class C, where C.fid = F.fid)
- (4) Select distinct F.fname from faculty F where 5 > (select count (E.Snum) from class C enabled, where C.name = E.name and C.fid = F.fid)
- (5) Select distinct s.sname from student s where student.snum not in (select E.num from enabled)
- (6) Select S.age, S.LVI from student s Group by S.age, S.level having S.level in (select S1.LVI from student S1 where S1.age = S.age
 group by S1.LVI, S1.age having count (*) > ALL (SELECT count * from student S2 where S1.age = S2.age Group by S2.LVI, S2.age));