Library DB

Library DB

BOOK

Bookid	Bookname	category

CUSTOMER

Custid	Custname

PURCHASE

Purchaseid	Custid	Bookingid	Purchasedate

Queries Used to CREATE and INSERT

```
    create table book(bookid int primary key,bookname vahrchar(20),category varchar(10));
    create table customer(custid int primary key,custname varchar(20));
    create table purchase(purchaseid int primary key,custid int,bookid int,purchasedate date,foreign key (custid) refereneces customer(custid), foreign key (bookid) references book (bookid));
```

BOOK

bookid	bookname	category
1001	abc	Horror
1002	qwe	Fantasy
1003	try	Scifi
1005	jee	Edu

CUSTOMER

custId	custname
2001	fgh
2002	ken
2003	felix
2004	Marzia
2005	Joel

PURCHASE

purchaseid	custid	bookid	purchasedate
3001	2002	1001	2000-04-07
3002	2004	1003	2000-05-17
3004	2005	1002	2000-11-18
3005	2001	1003	2000-11-17
3006	2003	1003	2000-11-17
3007	2003	1002	2000-05-17
3008	2003	1003	2000-04-18
3009	2002	1003	2000-04-18

WEBSITE USED FOR MYSQL

Link : http://sqlfiddle.com/

Queries

1. Identify the purchase details of the books that are puchased exactly on different dates by the same customer(s). Write an SQL query to display sutomer id and number of such puchases to be displayed as BOOKS for the identified purchase details.

```
select custid,count(*) as BOOKS from purchase
```

```
group by custid,purchasedate;
```

custid	BOOKS
2002	1
2004	1
2005	1
2001	1
2003	1
2003	1
2003	1
2002	1

---

2. Create a view to display the customer's id and title of the book for the purchase details of books, where the books of the same category are purchased by different customers on different dates.

```
select custid,bookname from purchase p,book b
where p.bookid= b.bookid
and category like 'Scifi'
group by custid,bookname;
```

custid	bookname
2004	try
2001	try
2003	try
2002	try

---

3. Create a new entity that contains the **Totalflightcharge** collected wrt each flight and demonstatrate trigger such that on new booking **Totalflightcharge** should be computed.

STEP 1 (Creating Entity)

```
create table totalfeecollected (pid int,totalfee int);
```

STEP 2 (Applying trigger)

```
    create trigger updatecount
    after insert on purchase
    for each row
    BEGIN

    DECLARE bso int;
    DECLARE bcat varchar(10);

    set bso = (select count(bookid) from purchase where purchaseid =
new.purchaseid);
    set bcat = (select category from purchase p,book b where p.bookid =
b.bookid and purchaseid = new.purchaseid);

    if not exists(select bookcategory from totalbook where
bookcategory=bcat)
    then
    insert into totalbook values(bso,bcat);

    else
    update totalbook set bookssold = bookssold+bso
    where bookcategory = bcat;

    end if;
    END$$


    delimiter $$
    create trigger feeupdate
    after insert on consultation
    for each row

    BEGIN

    DECLARE tpid int;
    DECLARE tfee int;

    set tpid = (select patientid from consultation where consultationid =
new.consultationid);
    set tfee = (select fee from consultation where consultationid =
```

```
new.consultationid);

    if not exists(select pid from totalfeecollected where pid = tpid)
    then
    insert into totalfeecollected values(tpid,tfee);

    else
    update totalfeecollected
    set totalfee = totalfee + tfee
    where pid = tpid;

    end if;
    END$$
```

---

4. Demonstrate the working procedure to diplay the customer's name and the title of the book for the purchase details of books, where the book is purchased on the same date exactly on the date **Suresh** has purchasde the book. **DO NOT DISPLAY** details of **Suresh** in the query.

```
    select custname,bookname from customer c,book b,purchase p
    where b.bookid = p.bookid and c.custid = p.custid
    and purchasedate in
    (select purchasedate from purchase p1,customer c1
    where p1.custid = c1.custid and custname like 'Marzia')
    and c.custname not like 'Marzia';

    // Here instead of Marzia Subsitute Suresh
```

custname	bookname
felix	qwe

---

5. Demonstrate the on **Delete/Update** cascade with simple query.

STEP 1 :

Get the **foreign key constraint name** using the below query.

```
show create table purchase; //Get the FK_ConstraintName
```

STEP 2:

Drop the constraint from the attribute that has foreign key constraint.

Has the foreign key contraint. We do this in order to add the cascade property to these attributes, as cascade is to be applied to Foriegn keys. We cannot update the foreign keys to cascade using alter, here we drop the constaint instead of the whole attribute and reassign the constraint to them with the cascade property.

We can drop the constraint with the below querry.

```
alter table purchase
delete constraint FK_book;
```

STEP 3:

We reassign the foreirgn key status to the attribute with the below querry.

```
alter table purchase
add constraint FK_book
foreign key(bookid) references book(bookid)
cascade on delete;
```

Table Before Delete

(Here flightid in booking is the foreign key referencing the flight table)

Flight

flightid	flightname	flighttype	sourcee	destination
2001	Air Asia	D	Bangalore	Mumbai

---

flightid	flightname	flighttype	sourcee	destination
2002	Indigo	D	Mumbai	Chennai
2003	Jet airways	I	Bangalore	Singapore
2004	Air Asia	D	Kelara	Hydrabad
2005	Vistara	I	Mumbai	US
2006	Goair	D	Chennai	Delhi
2007	Vistara	I	Bangalore	United States

Booking

bookingid	travelclass	flightcharge	bookingdate	flightid	custid
3001	Buisness	40000	2020-12-20	2002	1001
3002	Economy	30000	2020-04-19	2004	1003
3003	Economy	30000	2020-07-25	2001	1004
3004	Premium Eco	4000	2020-08-21	2005	1004
3005	Buisness	50000	2020-05-15	2005	1002
3006	Firstclass	70000	2020-07-17	2006	1005

STEP 4:

Delete a row from the table , having the foreign key of another table as the primary key, to demonstrate cascading.

```
delete from book where bookid = 1004;
```

Table After Delete

(Here flightid in booking is the foreign key referencing the flight table)

Flight

flightid	flightname	flighttype	sourcee	destination
2001	Air Asia	D	Bangalore	Mumbai

---

flightid	flightname	flighttype	sourcee	destination
2003	Jet airways	I	Bangalore	Singapore
2004	Air Asia	D	Kelara	Hydrabad
2005	Vistara	I	Mumbai	US
2006	Goair	D	Chennai	Delhi
2007	Vistara	I	Bangalore	United States

Booking

bookingid	travelclass	flightcharge	bookingdate	flightid	custid
3002	Economy	30000	2020-04-19	2004	1003
3003	Economy	30000	2020-07-25	2001	1004
3004	Premium Eco	4000	2020-08-21	2005	1004
3005	Buisness	50000	2020-05-15	2005	1002
3006	Firstclass	70000	2020-07-17	2006	1005

Note that the row with Flightid = 2002 has been deleted from Flight table , and so is deleted from the bookign table as well. This helps ensure that there is no redundancy and maintain consistency in the database.