



MULTIMEDIA UNIVERSITY ®

TIS1101 Database Fundamentals

**Title: Book Rental System**

**Date: 3rd February, 2016**

ASSIGNMENT CONTENTS

Business rules 1

ERD DIAGRAM 2

DATA dictionary 3-4

DAta definition command (DDL) 5-6

data insertion 7-12

***employee 7***

***customer 8***

***publisher 9***

***books 10***

***rental 11***

***rental details 12***

data manipulation 13-19

***“aggregate function” 13***

***“Group by” or “Having” clause 14***

***“nested query” or “sub-query” 15***

**VIEWS 16**

**TRIGGER 17**

**STORED PROCEDURE 18**

**ADDITIONAL QUERY 19**

1

**BUSINESS RULES**

1. **A customer makes many rental.**

**Each rental is made by only one customer.**

1. **One employee can check many rental.**

**A rental must be checked by only one employee.**

1. **Each rental has many rental details.**

**Every rental details is generated only for one rental.**

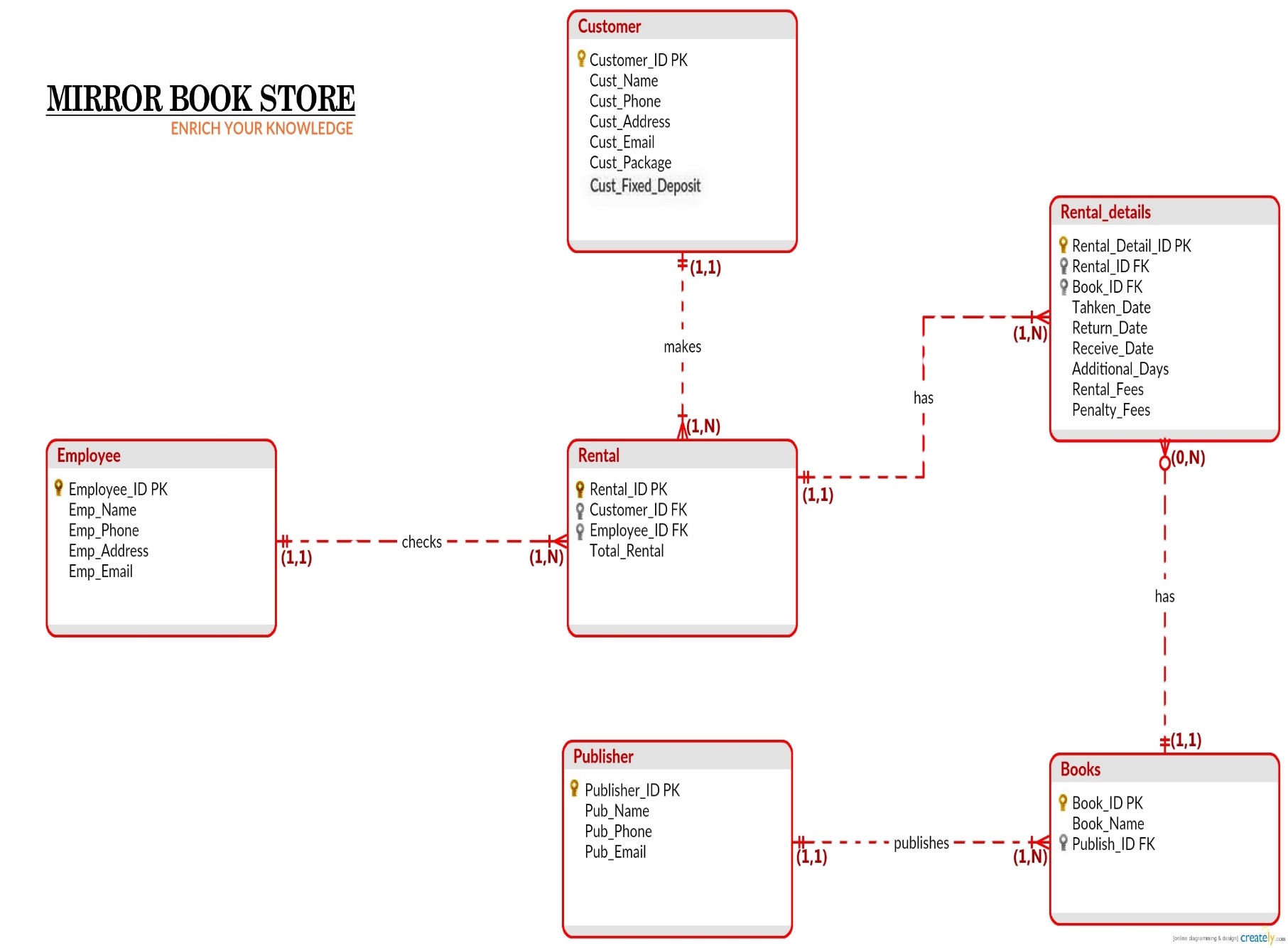
1. **Each book may or may not have many rental details.**

**A rental details must be generated for one book.**

1. **Each publisher has published at least one book.**

**Every book has been published by only one publisher.**

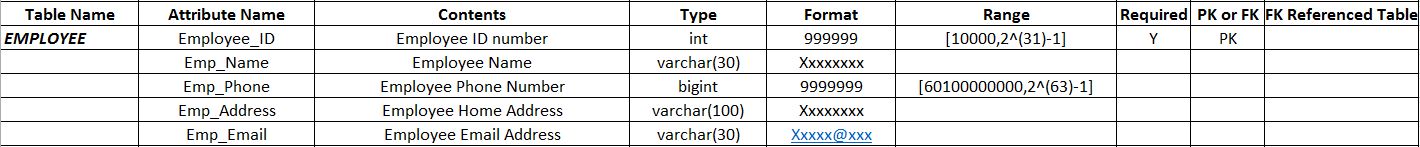
2

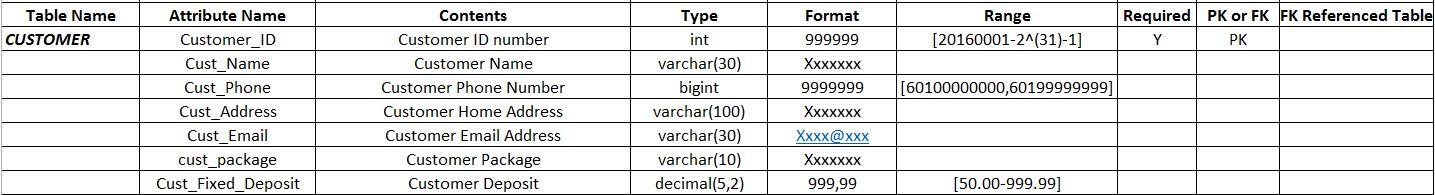
******

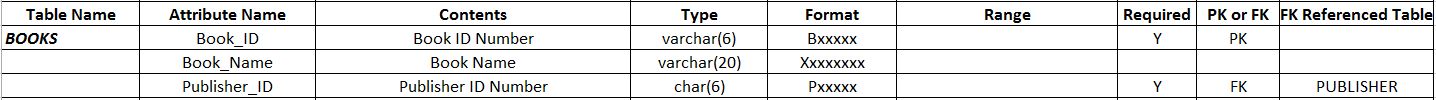
**3**

**DATA DICTIONARY**

**Employee:**

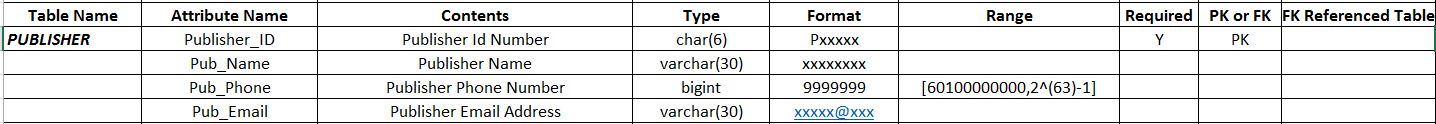


**Customer:**

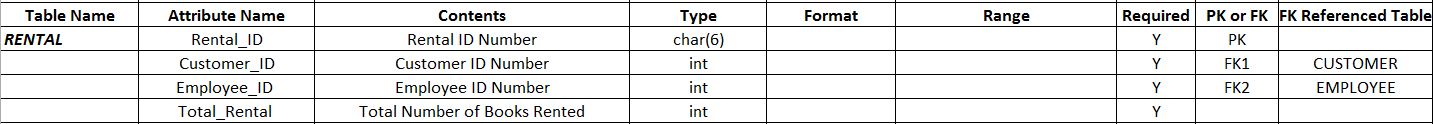
**Books:**

**Publisher:**

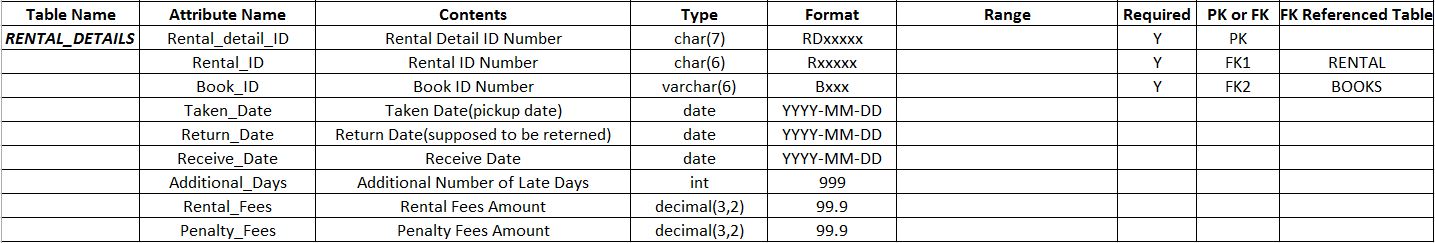
**4**

****

**Rental:**

****

**Rental Details:**

****

**5**

**DATA DEFINITION COMMAND**

**Create table Employee:**

create table Employee(

Employee\_ID int not null generated always as identity(start with 10000,increment by 1),

Emp\_Name varchar(30),

Emp\_Phone bigint,

Emp\_Address varchar(100),

Emp\_Email varchar(30),

Primary Key (Employee\_ID)

);

Employee table.PNG

**Create table Customer:**

create table Customer(

Customer\_ID int not null generated always as identity(start with 20160001, increment by 1),

Cust\_Name varchar(30),

Cust\_Phone bigint,

Cust\_Address varchar(100),

Cust\_Email varchar(30),

cust\_package varchar(10),

Cust\_Fixed\_Deposit decimal(5,2),

Primary Key (Customer\_ID)

);

Customer Table.PNG

**Create table Books:**

create table Books(

Book\_ID varchar(6) not null,

Book\_Name varchar(20),

Publisher\_ID char(6),

Primary Key(Book\_ID),

Foreign Key (Publisher\_ID) references Publisher on delete restrict

);

Books table.PNG

**6**

**Create table Publisher:**

create table Publisher(

Publisher\_ID char(6) not null,

Pub\_Name varchar(30),

Pub\_Phone bigint,

Pub\_Email varchar(30),

Primary Key (Publisher\_ID)

);

Publisher table.PNG

**Create table Rental:**

create table Rental(

Rental\_ID char(6) not null,

Customer\_ID int,

Employee\_ID int,

Total\_Rental int not null generated always as identity(start with 1,increment by 1),

Primary Key(Rental\_ID),

Foreign Key(Customer\_ID) references Customer on delete restrict,

Foreign Key(Employee\_ID) references Employee on delete restrict

);

Rental table.PNG

**Create table Rental\_Details:**

create table Rental\_Details(

Rental\_detail\_ID char(7) not null,

Rental\_ID char(6),

Book\_ID varchar(6),

Taken\_Date date,

Return\_Date date,

Receive\_Date date,

Additional\_Days int,

Rental\_Fees decimal(3,2),

Penalty\_Fees decimal(3,2),

Primary Key(Rental\_detail\_ID),

Foreign Key(Rental\_ID) references Rental on delete restrict,

Foreign Key(Book\_ID) references Books on delete restrict

);

Rental_Details table.PNG

**7**

**DATA INSERTION**

**Insertion of employee table:**

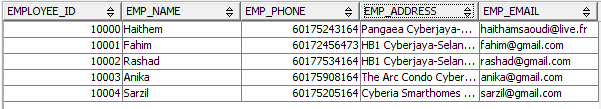
insert into Employee(Emp\_Name,Emp\_Phone,Emp\_Address,Emp\_Email) values('Haithem',60175243164,'Pangaea Cyberjaya-Selangor','haithamsaoudi@live.fr');

insert into Employee(Emp\_Name,Emp\_Phone,Emp\_Address,Emp\_Email) values('Fahim',60172456473,'HB1 Cyberjaya-Selangor','fahim@gmail.com');

insert into Employee(Emp\_Name,Emp\_Phone,Emp\_Address,Emp\_Email) values('Rashad',60177534164,'HB1 Cyberjaya-Selangor','rashad@gmail.com');

insert into Employee(Emp\_Name,Emp\_Phone,Emp\_Address,Emp\_Email) values('Anika',60175908164,'The Arc Condo Cyberjaya-Selangor','anika@gmail.com');

insert into Employee(Emp\_Name,Emp\_Phone,Emp\_Address,Emp\_Email) values('Sarzil',60175205164,'Cyberia Smarthomes Cyberjaya-Selangor','sarzil@gmail.com');



**8**

**Insertion of Customer table:**

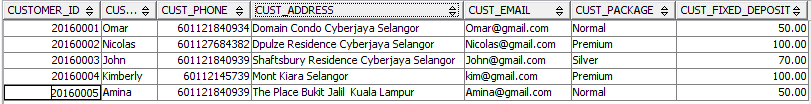
insert into Customer (Cust\_Name,Cust\_Phone,Cust\_Address,Cust\_Email ,cust\_package,Cust\_Fixed\_Deposit)values('Omar',601121840934,'Domain Condo Cyberjaya Selangor','Omar@gmail.com','Normal',50.00);

insert into Customer (Cust\_Name,Cust\_Phone,Cust\_Address,Cust\_Email ,cust\_package,Cust\_Fixed\_Deposit)values('Nicolas',601127684382,'Dpulze Residence Cyberjaya Selangor','Nicolas@gmail.com','Premium',100.00);

insert into Customer (Cust\_Name,Cust\_Phone,Cust\_Address,Cust\_Email ,cust\_package,Cust\_Fixed\_Deposit)values('John',601121840939,'Shaftsbury Residence Cyberjaya Selangor','John@gmail.com','Silver',70.00);

insert into Customer (Cust\_Name,Cust\_Phone,Cust\_Address,Cust\_Email ,cust\_package,Cust\_Fixed\_Deposit)values('Kimberly',60112145739,'Mont Kiara Selangor','kim@gmail.com','Premium',100.00);

insert into Customer (Cust\_Name,Cust\_Phone,Cust\_Address,Cust\_Email ,cust\_package,Cust\_Fixed\_Deposit)values('Amina',601121840939,'The Place Bukit Jalil  Kuala Lampur','Amina@gmail.com','Normal',50.00);



9

**Insertion of Publisher table:**

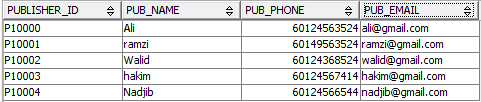
insert into Publisher values('P10000','Ali',60124563524,'ali@gmail.com');

insert into Publisher values('P10001','ramzi',60149563524,'ramzi@gmail.com');

insert into Publisher values('P10002','Walid',60124368524,'walid@gmail.com');

insert into Publisher values('P10003','hakim',60124567414,'hakim@gmail.com');

insert into Publisher values('P10004','Nadjib',60124566544,'nadjib@gmail.com');



10

**Insertion of Books table:**

insert into Books values('B1','Inferno','P10000');

insert into Books values('B2','Minds','P10001');

insert into Books values('B3','The Moon','P10002');

insert into Books values('B4','The Other World','P10003');

insert into Books values('B5','Discover The Truth','P10004');

insert into Books values('B6','Paradise','P10001');

insert into Books values('B7','Where is Alise','P10004');

insert into Books values('B8','Harry Poter','P10001');

insert into Books values('B9','The De Vinçi Code','P10000');

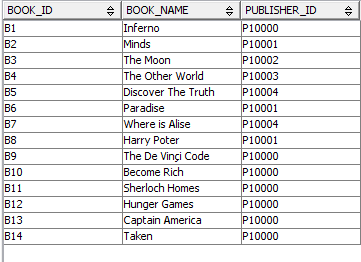
insert into Books values('B10','Become Rich','P10000');

insert into Books values('B11','Sherloch Homes','P10000');

insert into Books values('B12','Hunger Games','P10000');

insert into Books values('B13','Captain America','P10000');

insert into Books values('B14','Taken','P10000');



11

**Insertion of Rental table:**

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10000',20160001,10000);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10001',20160002,10000);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10002',20160003,10001);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10003',20160004,10000);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10004',20160005,10002);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10005',20160001,10003);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10006',20160005,10003);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10007',20160004,10000);

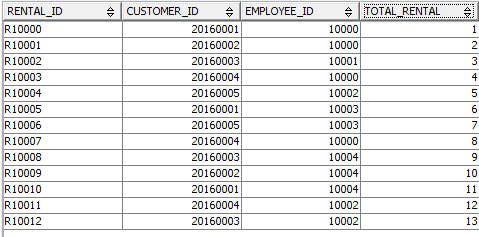
insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10008',20160003,10004);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10009',20160002,10004);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10010',20160001,10004);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10011',20160004,10002);

insert into Rental(Rental\_ID,Customer\_ID,Employee\_ID)values('R10012',20160003,10002);



12

**Insertion of Rental Details table:**

insert into Rental\_Details values('RD10000','R10000','B1','2015-01-01','2015-01-15','2015-01-28',NULL,2.50,NULL);

insert into Rental\_Details values('RD10001','R10001','B2','2015-08-09','2015-08-24','2015-08-17',NULL,2.00,NULL);

insert into Rental\_Details values('RD10002','R10002','B3','2015-12-01','2015-12-15','2015-12-04',NULL,1.70,NULL);

insert into Rental\_Details values('RD10003','R10003','B4','2015-11-10','2015-11-25','2015-11-28',NULL,3.50,NULL);

insert into Rental\_Details values('RD10004','R10004','B5','2015-09-11','2015-09-26','2015-09-28',NULL,1.50,NULL);

insert into Rental\_Details values('RD10005','R10005','B2','2015-05-12','2015-05-27','2015-05-28',NULL,2.00,NULL);

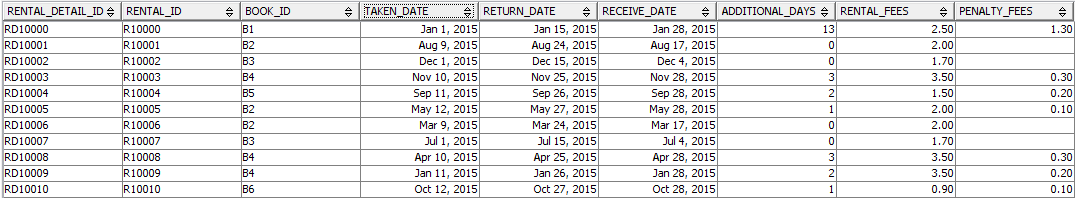
insert into Rental\_Details values('RD10006','R10006','B2','2015-03-09','2015-03-24','2015-03-17',NULL,2.00,NULL);

insert into Rental\_Details values('RD10007','R10007','B3','2015-07-01','2015-07-15','2015-07-04',NULL,1.70,NULL);

insert into Rental\_Details values('RD10008','R10008','B4','2015-04-10','2015-04-25','2015-04-28',NULL,3.50,NULL);

insert into Rental\_Details values('RD10009','R10009','B4','2015-01-11','2015-01-26','2015-01-28',NULL,3.50,NULL);

insert into Rental\_Details values('RD10010','R10010','B6','2015-10-12','2015-10-27','2015-10-28',NULL,0.90,NULL);



13

**DATA MANIPULATION COMMAND**

**AGGREGATE FUNCTION:**

Select Book\_Name,  COUNT(\*) as Total

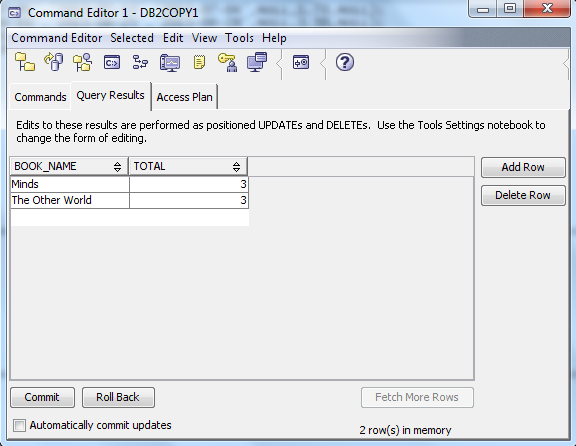
FROM Books b,Rental\_Details r

WHERE b.Book\_ID = r.Book\_ID

GROUP BY Book\_NAME

ORDER BY COUNT(\*) DESC

LIMIT 2;



This is an aggregate function which will show the most popular book in the library. It means which books are rented most of the time.

In our table “Minds” and “The Other World” books are rented 3 times by the customer and these are the highest amount of rented books.

So by this aggregate function we are showing the most popular books in the library.

14

**“GROUP BY”/ “HAVING” CLAUSE:**

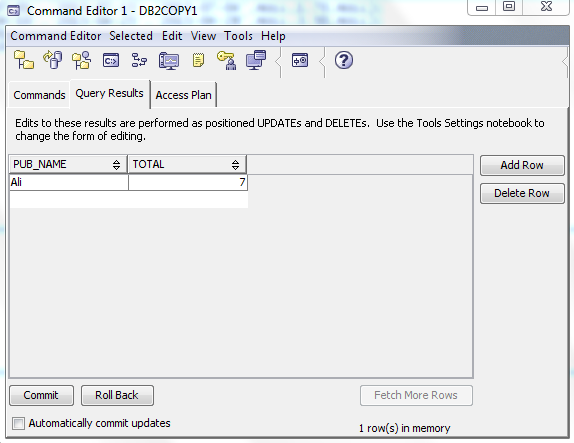
SELECT Pub\_Name, COUNT(\*) as Total

FROM Publisher p,Books b

WHERE b.Publisher\_ID = p.Publisher\_ID

GROUP BY Pub\_Name

HAVING COUNT(\*)>5;



This is a “having” clause function. By this function we are showing the PUBLISHER NAME who publish more than 5 books.

In our table Ali publisher publish 7 books which is more than 5.

15

**“NESTED QUERY” / “SUB-QUERY”:**

\*\*nested query 1

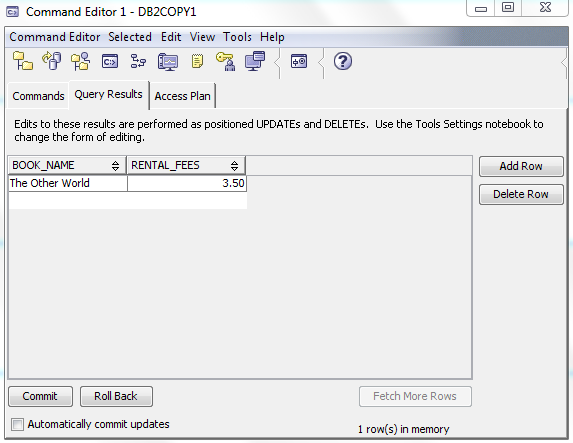
select Book\_Name,Rental\_Fees

from Rental\_Details r, Books b

Where b.Book\_ID=r.Book\_ID

And Rental\_Fees=(SELECT MAX(Rental\_Fees)from Rental\_Details)

Limit 1;



This is a “nested query”. By this function we can show the book name and the rental fees for the particular book which have the highest rental fees.

In our table “The Other World” has the highest rental fees.

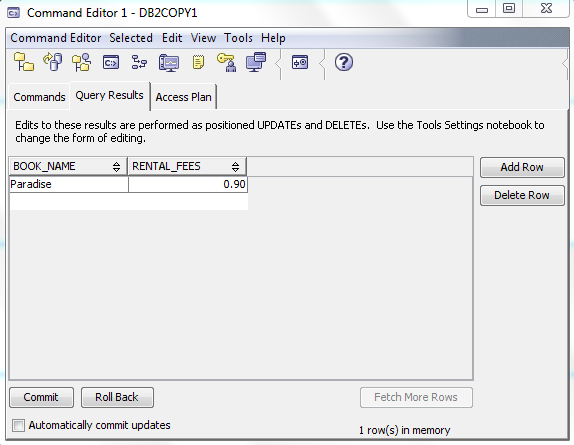
\*\* nested query 2

select Book\_Name,Rental\_Fees

from Rental\_Details r, Books b

Where b.Book\_ID=r.Book\_ID

And Rental\_Fees=(SELECT MIN(Rental\_Fees)from Rental\_Details);



This is a “nested query”. By this function we can show the book name and the rental fees for the particular book which have the lowest rental fees.

In our table “Paradise” has the highest rental fees

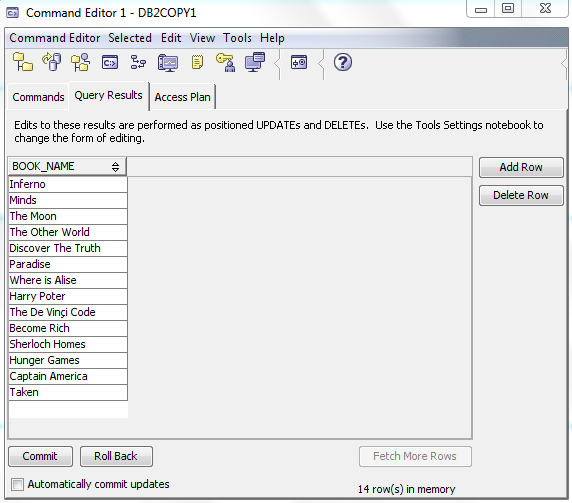
16

**VIEWS:**

\*\*view 1

create view Lib\_Books as

select Book\_Name from Books;



This is a “View”. By this function we can show all the book name in the library.

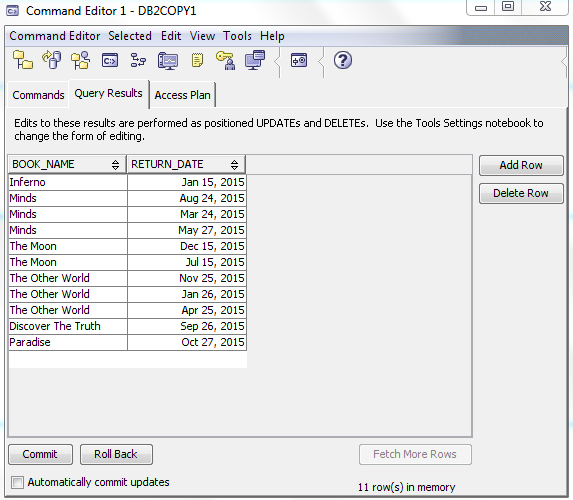
//view 2

create view Return\_Books as

select Book\_Name, Return\_Date

from Books b, Rental\_Details r

where b.Book\_ID=r.Book\_ID;



This is a “View”. By this function we can show all the books which in rented and the date when it will be returned in the library.

17

**TRIGGER:**

Create Trigger Rent\_det\_1

No cascade before insert on Rental\_Details

Referencing new as n

For each row mode db2sql

   Set n.Additional\_days=

   Case

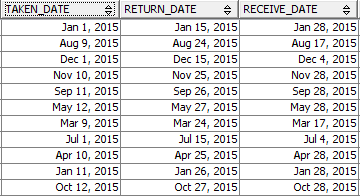
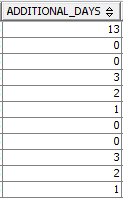
      When Receive\_Date > Return\_Date

        then Receive\_Date - Return\_Date

      When Receive\_Date <= Return\_Date

        then 0

 End;

This is a “Trigger”. By this function we can easy find out the additional days from the rental details table. If the customer return the books in late.

18

**STORED PROCEDURE:**

create procedure Prc\_Fees(IN charge\_per\_day decimal (3,2))

Begin

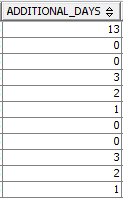
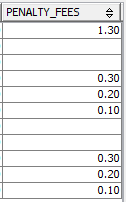
update Rental\_Details

set Penalty\_Fees = Additional\_Days \* charge\_per\_day

where Additional\_Days > 0;

END@

CALL Prc\_Fees(0.10)

---------------------------------------------------------------🡪 

This is a “Stored Procedure”. By this stored procedure library management can easily get the “Penalty Fees” for those customer who return the day late.

19

**ADDITIONAL QUERY:**

create table Employee(

Employee\_ID int not null generated always as identity(start with 10000,increment by 1),

Emp\_Name varchar(30),

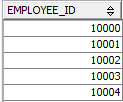
Emp\_Phone bigint,

Emp\_Address varchar(100),

Emp\_Email varchar(30),

Primary Key (Employee\_ID)

);



This is an “Additional Query”. By this function we just insert the employee name it will automatically give a employee ID increment by 1.