|  |
| --- |
| TE Comp Roll number : 8322 |
| **Assignment 1** Date of Submission : / / 2019 |
| Course outcomes: On successful completion of course learner will be able to:  **CSC502.1Understand the fundamentals of a database systems**  **CSC502.2. Design and draw ER and EER diagram for the real life problem**. |
| **Rubrics for assessment of Assignment 1:**   |  |  |  |  | | --- | --- | --- | --- | | **Indicator** | Excellent | Good | Average | | **Timeline**  **(3)** | submitted on time or early (3) | Submitted in same week (2) | Submitted in next week (1) | | **Understands the Problem(3)** | Identifies special factors that influences the approach before starting the problem | Understands the problem | Understands enough to solve part of the problem or to get part of the solution | | **Applies Appropriate Symbols to draw ER and EER Diagram (4)** | Explains how the Symbols are appropriate in ER and EER Diagram | Applies completely appropriate Symbols to draw ER and EER Diagram (2) | Applies some appropriate Symbols of ER to draw ER (1) | | **Timeline**  **(3)** | **Understands the Problem(3)** | **Applies Appropriate Symbols (4)** | **Total (10)** | |  |  |  |  | |
| **Teacher's Sign :** |

|  |
| --- |
| TE Comp Roll number : 8322 |
| **Assignment 2** Date of Submission : / / 2019 |
| **Course outcomes: On successful completion of course learner will be able to:**  **CSC502.3. Convert conceptual model to relational model and formulate relational algebra queries.**  **CSC502.4. Design and querying database using SQL.** |
| **Rubrics for assessment of Assignment 2:**   |  |  |  |  | | --- | --- | --- | --- | | **Indicator** | Excellent | Good | Average | | **Timeline**  **(2)** | submitted on time or early (2) | Submitted in same week (1) | Submitted in next week (0) | | **Understands the Relational algebra(4)** | Explains why procedures are appropriate for relational algebra(4) | Applies Appropriate Relational algebraic symbols for all queries (3) | Applies Appropriate Relational algebraic symbols <50% (2) | | **Applies Appropriate SQL (4)** | Explains why procedures are appropriate for SQL(4) | Applies Appropriate SQL (3) | Applies some appropriate SQL (2) | | **Timeline**  **(2)** | **Understands the Problem(3)** | **Applies Appropriate Procedures(4)** | **Total (10)** | |  |  |  |  | |
| **Teacher's Sign :** |

|  |
| --- |
| TE Comp Roll number : 8322 |
| **Assignment 3** Date of Submission : / / 2019 |
| **Course outcomes: On successful completion of course learner will be able to:**  CPC502.5: Analyze and apply concepts of normalization to relational database design. |
| **Rubrics for assessment of Assignment 3:**   |  |  |  |  | | --- | --- | --- | --- | | **Indicator** | Excellent | Good | Average | | **Timeline**  **(2)** | submitted on time or early (2) | Submitted in same week (1) | Submitted in next week (0) | | **Understands the Normalization types (4)** | Explains why Understands the Normalization types (4) | Understands the all types Normalization (3) | Understands some form of Normalization types < 50% (2) | | **Applies Appropriate Normalization (4)** | Explains why Normalization appropriate for given problem(4) | Applies Normalization appropriate for given problem (3) | Applies some part of Normalization appropriate for given problem (2) | | **Timeline**  **(2)** | **Understands the Normalization types (4))** | **Applies Appropriate Normalization (4)** | **Total (10)** | |  |  |  |  | |
| **Teacher's Sign :** |

|  |
| --- |
| TE Comp Roll number : 8322 |
| **Assignment 4**  Date of Submission : / / 2019 |
| **Course outcomes: On successful completion of course learner will be able to:**  CPC502.6 : Apply the concept of transaction, concurrency and recovery. |
| **Rubrics for assessment of Assignment 3:**   |  |  |  |  | | --- | --- | --- | --- | | **Indicator** | Excellent | Good | Average | | **Timeline**  **(2)** | submitted on time or early (2) | Submitted in same week (1) | Submitted in next week (0) | | **Understands the concept of transaction, concurrency and recovery with types (4)** | Explains why Understands the concepts (4) | Understands the all types (3) | Understands some form of (2) | | **Applies Appropriate concept for given Question. (4)** | Explains why appropriate concept for given problem(4) | Applies appropriate concept for given problem (3) | Applies some part of appropriate concept for given problem (2) | | **Timeline**  **(2)** | **Understands the concept of transaction, concurrency and recovery with types (4)** | **Applies Appropriate concept for given Question (4)** | **Total (10)** | |  |  |  |  | |
| **Teacher's Sign :** |

|  |
| --- |
| TE Computer Roll number : 8322 |
| **Mini project** Date of Submission : / / 2019 |
| **Course outcomes: On successful completion of course learner will be able to: ALL CO**  CPC502.1,2,3,4,5,6- All |
| **Rubrics for assessment of Mini project:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Indicator** | ***Poor*** | ***Average*** | ***Good*** | ***Excellent*** | | ***Timeline(3)*** | More than two sessions late (0) | Two sessions late (1) | One session late (2) | Early or on time (3) | | ***Implementation of client server Architecture (3)*** | No (0) | 2-T Architecture (1) | 3-T Architecture(2) | 3-T Architecture  With validation and verification  (3) | | ***Implementation of advanced SQL(4)*** | Nested queries and Complex queries (1) | Nested queries and Complex queries, with Views(2) | Nested queries and Complex queries, with Views and Triggers(3) | Views and Triggers, Functions , cursor and procedure(4) | |
| **Teacher's Sign :** |

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 1 Date of Implementation : / / 2019 |
| Related Course outcome : At the end of the course, Students will be able to design EER model and develop relational model |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness   * Maintains assignment deadline (3) | Assignment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness   * Complete all parts of ER diagram(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality   * Extent of plagiarism(2) | Copied it from someone else(0) | At least few questions have been done without copying(1) | Assignment has been solved completely without copying (2) | | Knowledge   * In depth knowledge of the assignment(2) | Unable to answer 2 questions(0) | Unable to answer 1 question (1) | Able to answer 2 questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 2 Date of Implementation : / / 2019 |
| Related Course outcome : At the end of the course, Students will be able to design EER model and develop relational model |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness   * Maintains assignment deadline (3) | Assignment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness   * Complete all parts of ER diagram(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality   * Extent of plagiarism(2) | Copied it from someone else(0) | At least few questions have been done without copying(1) | Assignment has been solved completely without copying (2) | | Knowledge   * In depth knowledge of the assignment(2) | Unable to answer 2 questions(0) | Unable to answer 1 question (1) | Able to answer 2 questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 3 Date of Implementation : / / 2019 |
| Aim : To implement data manipulation language (DML) commands |
| Tool Used : PostgreSQL |
| Related Course outcome : Students should be able to  Write queries in SQL to retrieve any type of information from a database. |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

**Task 1**

create table client\_master(

clientno varchar(6),

name varchar(20),

address varchar(30),

city varchar(15),

pincode numeric(8),

state varchar(15),

balance numeric(10,2)

)

create table Product\_master(

product\_no varchar(6),

description varchar(15),

Profit\_percent numeric(4,2),

Unit\_measure varchar(10),

Qty\_on\_hand numeric(8),

Reorder\_level numeric(8),

Sell\_price numeric(8,2),

Cost\_price numeric(8,2)

)

insert into client\_master values('123','abc','cba','a',10,'b',100)

insert into client\_master values('456','def','fed','c',20,'d',200)

insert into client\_master values('789','ghi','ihg','e',30,'f',300)

insert into client\_master values('012','jkl','lkj','g',40,'h',400)

insert into client\_master values('345','mno','onm','i',50,'j',500)

insert into client\_master values('345','mno','onm','mumbai',50,'j',500)

insert into Product\_master values('567','hello',50,'unit',60,70,200,100)

insert into Product\_master values('890','world',60,'unit2',70,80,300,200)

insert into Product\_master values('678','hi',70,'unit3',80,90,400,300)

insert into Product\_master values('645','bye',80,'unit4',90,100,500,400)

insert into Product\_master values('876','goodbye',90,'unit5',100,110,600,500)

select name from client\_master

select \* from client\_master

select name,city from client\_master

select \* from Product\_master

select \* from client\_master where city='mumbai'

update client\_master set city='mumbai' where clientno='456'

update client\_master set balance=1000 where clientno='456'

update Product\_master set Cost\_price=560 where description='hi'

delete from Product\_master where Qty\_on\_hand<70

delete from client\_master where state='d'

select \* from client\_master

select \* from Product\_master

**Task 2**

create table Library(

material varchar(60), countmat int

)

insert into Library values("Books",50)

insert into Library values("CD's",20)

insert into Library values("Magazines",20)

select \* from Library

create table Member(

name varchar(50), memid numeric(10),

dept varchar(20)

)

insert into Member values("Abc",210,"Computers")

insert into Member values("Def",211,"Electronics")

insert into Member values("Ghi",212,"Info.Tech")

select \* from Member

create table Student(

sid numeric(10), rollno numeric(10),

class varchar(20), course varchar(20)

)

insert into Student values(188,12,"TE Computers","Computers")

insert into Student values(189,13,"TE Computers","Computers")

insert into Student values(190,14,"TE Computers","Computers")

select \* from Student

create table Teacher(

tid numeric(10), enrollno numeric(10),

dept varchar(20)

)

insert into Teacher values(121,13423,"Computers")

insert into Teacher values(122,13424,"Info.Tech")

insert into Teacher values(123,13425,"Production")

select \* from Teacher

create table Staff(

stid numeric(10), dept varchar(50)

)

insert into Staff values(131,"Computers")

insert into Staff values(132,"Info.Tech")

select \* from Staff

create table Material(

mid numeric(10),

type varchar(50),

qty int

)

insert into Material values(141,"Books",19)

insert into Material values(142,"Cds",16)

insert into Material values(143,"Magazines",11)

select \* from Material

create table Books(

bid numeric(10), bname varchar(50),

qty int, publisher varchar(20)

)

insert into Books values(45,"DBMS",5,"Techmax")

insert into Books values(46,"CN",8,"Technical")

insert into Books values(47,"AA",6,"Coreman")

select \* from Books

create table CDs(

cid numeric(10), cdname varchar(50),

qty int, publisher varchar(20)

)

insert into CDs values(55,"AWS",9,"Amazon")

insert into CDs values(56,"Servlet",4,"Java")

insert into CDs values(57,"OCPJP",3,"Oracle")

select \* from CDs

create table Magazines(

mid numeric(10),

magname varchar(50),

qty int,

publisher varchar(20)

}

insert into Magazines values(65,"Fragmag1",7,"Abc")

insert into Magazines values(66,"Fragmag2",3,"Pqr")

insert into Magazines values(67,"Fragmag3",1,"Lmn")

select \* from Magazines

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 4 Date of Implementation : / / 2019 |
| Aim : To implement simple SQL commands |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

create table sales\_order(

name varchar(20), order\_no varchar(6), order\_date date,

client\_no varchar(6), delv\_addr varchar(25), salesman\_no varchar(6),

delv\_type varchar(20), billed\_yn varchar(20), delv\_date date,

order\_status varchar(20), daysreqdfordelivery int, city varchar(20),

bal\_due int, sellingprice int

)

insert into sales\_order values('Ram','123', '2018-01-12','C001','abc','20', 'Prepaid','Invoice Given','2018-01-17','delivered',5,'Mumbai',12000,15000)

insert into sales\_order values('Raj','456', '2018-02-18','C002','def','21', 'COD','Invoice Not Given','2018-02-20','delivered',2,'Pune',8000,9500)

insert into sales\_order values('Rohit','789', '2018-05-22','C003','ghi','22', 'Prepaid','Invoice Given','2018-05-24','out for delivery',2,'Bengaluru',14000,16000)

insert into sales\_order values('Vedant','012', '2018-07-24','C004','jkl','23', 'Prepaid','Invoice Not Given','2018-07-27','shipping',3,'Delhi',3000,3500)

insert into sales\_order values('Caje','345', '2018-09-26','C005','mno','24', 'COD','Invoice Given','2018-09-30','order placed',4,'Hyderabad',2500,4500)

insert into sales\_order values('Chinmay','678', '2018-09-27','C001','pqr','25', 'COD','Invoice Given','2018-11-30','order placed',3,'Kolkata',1000,1200)

select \* from sales\_order

select \* from sales\_order where name like '\_a%'

--4

select \* from sales\_order where city='Mumbai'

--5

select \* from sales\_order where bal\_due>=10000

--6

select from sales\_order where extract('month' from delv\_date)=01

--7

select \* from sales\_order where client\_no='C001' or client\_no='C002'

--8

select \* from sales\_order where sellingprice>2000 and sellingprice<=5000

--9

select order\_no,sellingprice\*1.5 as "new\_price" from sales\_order where sellingprice>1500

--10

select count(\*) from sales\_order

--11

select avg(sellingprice) from sales\_order

--12

select min(sellingprice),max(sellingprice) from sales\_order

--13

select count(\*) from sales\_order where sellingprice>1500

--14

select order\_no,extract(from order\_date)from sales\_order

--15

select to\_char(order\_date,'dd-mm-yyyy') from sales\_order

--16

select delv\_date as "Delivery Date",to\_char(month from delv\_date,'day')as "Month" from sales\_order

--17

select now()::date + 15

--18

select extract('day' from age(now()::date,delv\_date)) from sales\_order

--19

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 5(Part-1) Date of Implementation : / / 2019 |
| Aim : To implement data definition language (DDL) commands |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 5(Part-2) Date of Implementation : / / 2019 |
| Aim : To implement Integrity constraints |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

--1

create table company1(

id int , name text not null, age int not null, address varchar(50), salary real

);

select \* from company1

--2

insert into company1 values(01,NULL,NULL,'LA',250000)

--3

create table company3(

id int not null ,

name text,

age int unique,

address varchar(50),

salary real

)

--4

insert into company3 values(02,'CAJE',25,'NY',50000)

insert into company3 values(02,'CAJE',25,'NY',50000)

--5

create table company4(

id int primary key, name text, age int, address varchar(50), salary real

)

--6

insert into company4 values(03,'LOGAN',11,'CHICAGO',40000)

insert into company4 values(03,'LOGAN',11,'CHICAGO',40000)

--7

create table employee(

id int primary key, name text, age int, address varchar(50), salary real

)

create table department(

id int primary key,

name varchar(50) not null,

empid int references employee(id)

)

--8

insert into employee values(1,'JAKE',11,'CHICAGO',40000)

insert into employee values(2,'SENSEI',11,'US',40000)

--9

select \* from employee

insert into department values(07,'AJ STYLES',1)

insert into department values(08,'JOHN',2)

--10

create table company5(

id int, name text, age int, address varchar(50),

salary real check(salary>0)

)

insert into company5 values(09,'BAREN',24,'MADH',-1)

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 6 Date of Implementation : / / 2019 |
| Aim : To implement Join and complex SQL commands |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

--1

create table sales\_order\_details(

order\_no varchar(6) primary key,

product\_no varchar(6),

qty\_ordered numeric(8),

qty\_disp numeric(8),

product\_rate numeric(10,2),

order\_date date not null

)

drop table sales\_order\_details

create table customer(

cid integer primary key,

cname varchar(25),

address varchar(25),

pno numeric(10)

)

drop table customer

create table cust\_order(

cid int references customer(cid),

order\_no varchar(6) references sales\_order\_details(order\_no)

)

drop table cust\_order

--2

insert into sales\_order\_details values('123','321',5,3,50000,'2018-01-12')

insert into sales\_order\_details values('456','654',4,2,40000,'2018-03-15')

insert into sales\_order\_details values('789','987',3,1,55000,'2018-04-13')

insert into sales\_order\_details values('012','210',6,3,75000,'2018-02-14')

insert into sales\_order\_details values('345','543',7,5,45000,'2018-01-28')

insert into customer values(1,'Abc','Cba','1234567890')

insert into customer values(2,'Def','Fed','1234554321')

insert into customer values(3,'Ghi','Ihg','5432112345')

insert into customer values(4,'Jkl','Lkj','9876543210')

insert into cust\_order values(1,'123')

insert into cust\_order values(2,'456')

insert into cust\_order values(3,'789')

insert into cust\_order values(4,'012')

--3

select product\_no, qty\_ordered from customer c cross join cust\_order co cross join sales\_order\_details sod

where c.cid=co.cid and co.order\_no=sod.order\_no

--4

select product\_rate from customer c,cust\_orderco,sales\_order\_details sod

where c.cid=co.cid and co.order\_no=sod.order\_no

--5

select avg(qty\_ordered) from customer c,cust\_orderco,sales\_order\_details sod

where c.cid=co.cid and co.order\_no=sod.order\_no and sod.product\_rate<=15000

--6

select sum(product\_rate) from customer c,cust\_orderco,sales\_order\_details sod

where c.cid=co.cid and co.order\_no=sod.order\_no and extract('month' from order\_date)=01

--7

select qty\_ordered from customer c,cust\_orderco,sales\_order\_details sod

where c.cid=co.cid and co.order\_no=sod.order\_no and sod.qty\_ordered>10

--8

select cname , Y.max from customer cu,(

select max(X.tq) as "max" from (

select c.cname , sum(Qty\_ordered) as "tq" from customer c cross join cust\_order co cross join sales\_order\_details sod

where c.cid = co.cid and co.order\_no = sod.order\_no

group by c.cname ) as X ) as Y LIMIT 1

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 7 Date of Implementation : / / 2019 |
| Aim : To implement Nested Sub-queries in SQL |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

--1

SELECT product\_no , qty\_desc

FROM sales\_order\_details

WHERE NOT EXISTS

(SELECT \* FROM cust\_order

WHERE sales\_order\_details.order\_no = cust\_order.order\_no);

--2

SELECT cname , address

FROM customer cross join cust\_order cross join sales\_order\_details

WHERE customer.cid = cust\_order.cid AND cust\_order.order\_no = sales\_order\_details.order\_no AND cust\_order.order\_no = '101' ;

--3

SELECT distinct cname

FROM customer cross join cust\_order cross join sales\_order\_details

WHERE customer.cid = cust\_order.cid AND cust\_order.order\_no = sales\_order\_details.order\_no

anddate\_part('month', sales\_order\_details.order\_date) <= 5;

--4

SELECT customer.cid , cname

FROM customer cross join cust\_order cross join sales\_order\_details

WHERE customer.cid = cust\_order.cid AND cust\_order.order\_no = sales\_order\_details.order\_no

AND cust\_order.order\_no = '103'

--5

SELECT cname

FROM customer cross join cust\_order cross join sales\_order\_details

WHERE customer.cid = cust\_order.cid AND cust\_order.order\_no = sales\_order\_details.order\_no

AND sales\_order\_details.product\_rate> 50000

--6

SELECT cname , sales\_order\_details.qty\_desc , product\_rate

FROM customer cross join cust\_order cross join sales\_order\_details

WHERE customer.cid = cust\_order.cid AND cust\_order.order\_no = sales\_order\_details.order\_no

AND customer.cname = 'Logan'

--7

SELECT product\_no FROM sales\_order\_details

WHERE NOT EXISTS

(SELECT \* FROM cust\_order

WHERE sales\_order\_details.order\_no = cust\_order.order\_no);

--8

SELECT X.product\_no , X.count , qty\_desc FROM sales\_order\_details ,(

SELECT sales\_order\_details.product\_no as "product\_no", sum(qty\_ordered) AS "count"

FROM customer cross join cust\_order cross join sales\_order\_details

WHERE customer.cid = cust\_order.cid AND cust\_order.order\_no = sales\_order\_details.order\_no

GROUP BY sales\_order\_details.product\_no

) AS X

WHERE X.product\_no = sales\_order\_details.product\_no

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 8 Date of Implementation : / / 2019 |
| Aim : To implement PL/pgSQL |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

**1. Write a block to display sum of digits of a three digit number**

**Code :(Using PL/SQL)**

do $$

<<block1>>

Declare

n int:=534;

n1 int;n2 int;n3 int;ans int;

Begin

n1=n/100;

n2=(n/10)%10;

n3=n%10;

ans=n1+n2+n3;

Raise Notice 'SUM OF 3 DIGITS OF A 3 DIGIT NUMBER IS=% ',ans;

end block1 $$;

**Output :**

NOTICE: SUM OF 3 DIGITS OF A 3 DIGIT NUMBER IS=12

Query returned successfully with no result in 12 ms.

**2. Write a block to display square of 1 to 10**

**Code :(Using PL/SQL)**

do $$

<<block2>>

Declare

n int:=1;

Begin

while n<=10

loop

Raise Notice 'SQUARE OF % number =% ',n,(n\*n);

n:=n+1;

end loop;

end block2 $$;

**Output :**

NOTICE: SQUARE OF 1 number =1

NOTICE: SQUARE OF 2 number =4

NOTICE: SQUARE OF 3 number =9

NOTICE: SQUARE OF 4 number =16

NOTICE: SQUARE OF 5 number =25

NOTICE: SQUARE OF 6 number =36

NOTICE: SQUARE OF 7 number =49

NOTICE: SQUARE OF 8 number =64

NOTICE: SQUARE OF 9 number =81

NOTICE: SQUARE OF 10 number =100

Query returned successfully with no result in 12 ms.

**3. Write a block to display Fibonacci series upto 8th term (start with 0,1)**

**Code :(Using Function)**

create function f1() returns varchar(50)

as $$

<<bl>>

declare

count int:=3; f int; s int;t int;

begin

f=0;

s=1;

Raise Notice 'The Term 1 of Fibonacci series is %',f;

Raise Notice 'The Term 2 of Fibonacci series is %',s;

t=f+s;

Raise Notice 'The Term % of Fibonacci series is %',count,t;

count=count+1;

while count<=8

loop

f=s;

s=t;

t=f+s;

Raise Notice 'The Term % of Fibonacci series is %',count,t;

count:=count+1;

end loop;

return 'Successful';

end;

$$ language plpgsql

select f1();

**Output :**

NOTICE: The Term 1 of Fibonacci series is 0

NOTICE: The Term 2 of Fibonacci series is 1

NOTICE: The Term 3 of Fibonacci series is 1

NOTICE: The Term 4 of Fibonacci series is 2

NOTICE: The Term 5 of Fibonacci series is 3

NOTICE: The Term 6 of Fibonacci series is 5

NOTICE: The Term 7 of Fibonacci series is 8

NOTICE: The Term 8 of Fibonacci series is 13

Total query runtime: 16 ms.

1 row retrieved.

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 9 Date of Implementation : / / 2019 |
| Aim : To implement Functions and Triggers |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use SQL : Standard language of relational database |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

**--1**

create or replace function f1() returns integer

as $$

<<bl>>

declare

fact int:=1; n int:=4;

begin

while n>0

loop

fact=fact\*n;

n=n-1;

end loop;

return fact;

end;

$$ language plpgsql

drop function f1();

select f1() AS "Factorial :";

**--2**

create table emp(

id integer,

name varchar(100),

salary integer

);

insert into emp values(1,'Logan',1200000);

insert into emp values(2,'Jake',450000);

insert into emp values(3,'Tom',80500);

**--3**

CREATE OR REPLACE FUNCTION findAverage ()

RETURNS decimal AS $avgSalary$

declare

average decimal;

BEGIN

SELECT avg(salary) into average FROM emp;

RETURN average;

END;

$avgSalary$ LANGUAGE plpgsql;

select findAverage();

**--4**

select \* from emp

CREATE TRIGGER beforeInsert BEFORE INSERT ON emp

FOR EACH ROW EXECUTE PROCEDURE executeTrigger();

CREATE OR REPLACE FUNCTION executeTrigger() RETURNS TRIGGER AS $empTable$

BEGIN

raise notice 'Not Allowed';

return new;

END;

$empTable$ LANGUAGE plpgsql;

insert into emp values(4,'Stark',4500); --NOTICE: Not Allowed

CREATE TRIGGER beforeUpdate BEFORE UPDATE ON emp

FOR EACH ROW EXECUTE PROCEDURE executeTrigger();

update emp set name ='Pew' where id=4; --NOTICE: Not Allowed

CREATE TRIGGER beforeDelete BEFORE DELETE ON emp

FOR EACH ROW EXECUTE PROCEDURE executeTrigger();

delete from emp where id=4; --NOTICE: Not Allowed

**--5**

CREATE TRIGGER afterInsert AFTER INSERT ON emp

FOR EACH ROW EXECUTE PROCEDURE executeTrigger();

CREATE OR REPLACE FUNCTION executeTrigger() RETURNS TRIGGER AS $empTable$

BEGIN

INSERT INTO AUDIT(EMP\_ID, ENTRY\_DATE) VALUES (new.ID, current\_timestamp);

return new;

END;

$empTable$ LANGUAGE plpgsql;

insert into emp values(5,'Malone',4500);

select \* from audit;

CREATE TRIGGER afterUpdate AFTER UPDATE ON emp

FOR EACH ROW EXECUTE PROCEDURE executeTrigger();

update emp set name ='Pew' where id=4;

CREATE TRIGGER afterDelete AFTER DELETE ON emp

FOR EACH ROW EXECUTE PROCEDURE executeTrigger();

delete from emp where id=4; --NOTICE: Not Allowed

|  |
| --- |
| TE Computers Roll number : 8322 |
| Experiment no. : 10 Date of Implementation : / / 2019 |
| Aim : Simple Transaction implementation |
| Tool Used : PostgreSQL |
| Related Course outcome : At the end of the course, Students will be able to Use and Apply the concept of transaction, concurrency and recovery |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness  Maintains Experiment deadline (3) | Experiment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness  Complete all parts of Experiment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality  Extent of plagiarism(2) | Copied it from someone else(0) | At least try to implement but could not succeed (1) | Implemented (2) | | Knowledge  In depth knowledge of the Experiment(2) | Unable to answer any questions(0) | Unable to answer few questions (1) | Able to answer all questions (2) | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | |
| **Total : (Out of 10)** |
| **Teacher's Sign :** |

public class Actor {

    private String firstName;

    private String lastName;

    public Actor(String firstName, String lastName) {

        this.firstName = firstName;

        this.lastName = lastName;

    }

    public Actor() {    }

    public String getFirstName() {return firstName;}

    public void setFirstName(String firstName) { this.firstName = firstName; }

    public String getLastName() { return lastName; }

    public void setLastName(String lastName) { this.lastName = lastName; }

}

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class App {

    public Connection connect() throws SQLException {

        return DriverManager.getConnection(“jdbc:postgresql://localhost/dvdrental”, “postgres:, “postgres”);

    }

    private App close(AutoCloseable closeable) {

        try {

            if (closeable != null)

                closeable.close();

        } catch (Exception e) {

            System.out.println(e.getMessage());

        }

        return this;

    }

    public void addActorAndAssignFilm(Actor actor, int filmId) {

        Connection conn = null;

        PreparedStatement pstmt = null;

        PreparedStatement pstmt2 = null;

        ResultSet rs = null;

        // insert an actor into the actor table

        String SQLInsertActor = "INSERT INTO actor(first\_name,last\_name) "

                + "VALUES(?,?)";

        // assign actor to a film

        String SQLAssignActor = "INSERT INTO film\_actor(actor\_id,film\_id) "

                + "VALUES(?,?)";

        int actorId = 0;

        try {

            // connect to the database

            conn = connect();

            conn.setAutoCommit(false);

            // add actor

            pstmt = conn.prepareStatement(SQLInsertActor,

                    Statement.RETURN\_GENERATED\_KEYS);

            pstmt.setString(1, actor.getFirstName());

            pstmt.setString(2, actor.getLastName());

            int affectedRows = pstmt.executeUpdate();

            if (affectedRows > 0) {

                // get actor id

                rs = pstmt.getGeneratedKeys();

                if (rs.next()) {

                    actorId = rs.getInt(1);

                    if (actorId > 0) {

                        pstmt2 = conn.prepareStatement(SQLAssignActor);

                        pstmt2.setInt(1, actorId);

                        pstmt2.setInt(2, filmId);

                        pstmt2.executeUpdate();

                    }

                }

            } else

                // rollback the transaction if the insert failed

                conn.rollback();

            // commit the transaction if everything is fine

            conn.commit();

            System.out.println(String.format("The actor was inserted with id %d and "

                            + "assigned to the film %d", actorId, filmId));

        } catch (SQLException ex) {

            System.out.println(ex.getMessage());

            // roll back the transaction

            System.out.println("Rolling back the transaction...");

            try {

                if (conn != null)   conn.rollback();

            } catch (SQLException e) {}

        } finally {

            this.close(rs)

                    .close(pstmt)

                    .close(pstmt2)

                    .close(conn);

        }

    }

    public static void main(String[] args) {

        App app = new App();

        // OK transaction

         app.addActorAndAssignFilm(new Actor("Bruce", "Lee"), 1);

        // Failed transaction

        // app.addActorAndAssignFilm(new Actor("Lily", "Lee"), 9999);

    }

}