database ques ans

Final examination, Fall 2018

1.b)

CREATE DATABASE practice

CREATE TABLE buyer(

bid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

bname VARCHAR(100),

address VARCHAR(100))

INSERT INTO buyer(bname,address)VALUES('Ask','Dhaka'),('csk','Rangpur'),('cisalpha','Dinajpur')

UPDATE buyer SET bname='ashik' WHERE bid=1;

UPDATE buyer SET bname='sos' WHERE bid=2;

UPDATE buyer SET bname='zhorna' WHERE bid=3;

CREATE TABLE factory(

fid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

fname VARCHAR(100),

address VARCHAR(100))

INSERT INTO factory(fname,address)VALUES('fAk','Dhaka'),('dak','Rangpur'),('tak','Dinajpur')

CREATE TABLE buyer\_order(

bpo\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

product VARCHAR(100),

color VARCHAR(100),

qty INT,

bprice INT,

bid INT,

FOREIGN KEY buyer\_order\_bid(bid) REFERENCES buyer(bid))

INSERT INTO buyer\_order(product,color,qty,bprice,bid)VALUES('pen','black',10,25000,1),

('pencil','white',20,100000,2),('rubber','yellow',15,15000,1),('craser','black',25,15000,3),

('khata','white',5,25000,1)

CREATE TABLE factory\_order(

fpo\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

qty INT,

fprice DOUBLE,

bpo\_id INT,

fid INT,

FOREIGN KEY buyer\_order\_bid1(bpo\_id) REFERENCES buyer\_order(bpo\_id ),

FOREIGN KEY buyer\_order\_bid2(fid) REFERENCES factory(fid))

INSERT INTO factory\_order(qty,fprice,bpo\_id,fid)VALUES(10,60000,1,2),(20,40000,2,3),(30,55000,3,2),

(40,30000,4,1),(50,45000,5,2)

update factory\_order set bpo\_id=4 where fpo\_id=1

1. SELECT bname,address FROM buyers
2. select fname from factory inner join factory\_order on (factory.fid=factory\_order.fid)

inner join buyer\_order on (factory\_order.bpo\_id=buyer\_order.bpo\_id)

inner join buyer on (buyer\_order.bid=buyer.bid) where bname='csk'

1. select fpo\_id from factory\_order inner join buyer\_order on(factory\_order.bpo\_id=buyer\_order.bpo\_id)

inner join buyer on buyer\_order.bid=buyer.bid where buyer.bname='cisalpha' having MAX(factory\_order.fprice)

1. SELECT count(factory.fid) AS no\_fact

FROM factory

INNER JOIN factory\_order ON (factory.fid=factory\_order.fid)

INNER JOIN buyer\_order ON (factory\_order.bpo\_id=buyer\_order.bpo\_id)

GROUP BY factory.fid

HAVING SUM(buyer\_order.bprice) > 50000

1. select bpo\_id from buyer\_order inner join buyer on buyer.bid=buyer\_order.bid where bname LIKE 'a\_%\_%\_%' OR bname LIKE 's\_%\_%\_%' OR bname LIKE 'z\_%\_%\_%'
2. CREATE INDEX buyer\_order\_index ON buyer\_order(product,color)

4.b)

CREATE TABLE customer(

cust\_no INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

cust\_name VARCHAR(100),

address VARCHAR(100))

INSERT INTO customer(cust\_name,address)VALUES('karim','dhaka'),('abdul','rangpur'),('rahim','dinajpur')

CREATE TABLE sales\_person(

sid INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

sname VARCHAR(100))

INSERT INTO sales\_person(sname)VALUES('ask'),('bsk'),('csk')

CREATE TABLE orders(

orders\_no INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

cust\_no INT,

sid INT,

qty INT,

STATUS VARCHAR(100),

date\_order DATE,

FOREIGN KEY orders\_1(cust\_no) REFERENCES customer(cust\_no),

FOREIGN KEY orders\_2(sid) REFERENCES sales\_person(sid))

INSERT INTO orders(cust\_no,sid,qty,STATUS,date\_order)VALUES(2,1,10,'canceled','2020-02-01'),

(1,2,20,'resolved','2020-02-03'),(3,2,15,'shipped','2020-02-04'),(2,3,10,'canceled','2020-02-05'),

(2,1,5,'disputed','2020-02-06'),(1,3,6,'resloved','2020-02-07'),(2,1,9,'canceled','2020-02-08'),

(2,1,11,'shipped','2020-02-09'),(2,1,11,'resolved','2020-02-09'),(3,2,11,'canceled','2020-02-09')

DELIMITER $$

CREATE PROCEDURE order\_proc(IN c\_name VARCHAR(20),IN s\_name VARCHAR(20))

BEGIN

SELECT COUNT(\*) AS Shipped FROM orders INNER JOIN customer ON customer.cust\_no=orders.cust\_no

INNER JOIN sales\_person ON sales\_person.sid=orders.sid WHERE orders.status='shipped' AND

customer.cust\_name=c\_name AND sales\_person.sname=s\_name;

SELECT COUNT(\*) AS Canceled FROM orders INNER JOIN customer ON customer.cust\_no=orders.cust\_no

INNER JOIN sales\_person ON sales\_person.sid=orders.sid WHERE orders.status='canceled' AND

customer.cust\_name=c\_name AND sales\_person.sname=s\_name;

SELECT COUNT(\*) AS resolved FROM orders INNER JOIN customer ON customer.cust\_no=orders.cust\_no

INNER JOIN sales\_person ON sales\_person.sid=orders.sid WHERE orders.status='resolved' AND

customer.cust\_name=c\_name AND sales\_person.sname=s\_name;

SELECT COUNT(\*) AS Disputed FROM orders INNER JOIN customer ON customer.cust\_no=orders.cust\_no

INNER JOIN sales\_person ON sales\_person.sid=orders.sid WHERE orders.status='disputed' AND

customer.cust\_name=c\_name AND sales\_person.sname=s\_name;

END $$

DELIMITER ;

CALL order\_proc('abdul','ask')

6.a)

CREATE TABLE Student\_marks(student\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,NAME VARCHAR(20),

sub1 INT,sub2 INT,sub3 INT,sub4 INT,sub5 INT)

INSERT INTO Student\_marks(NAME,sub1,sub2,sub3,sub4,sub5)VALUES('rahim',65,67,50,59,67),('karim',95,96,97,99,100),

('abdul',76,78,75,94,62),('rahman',54,62,46,33,32),('raju',45,36,35,39,38),('kanon',45,55,68,65,42)

CREATE TABLE Grades(Student\_id INT,average\_marks FLOAT,grade VARCHAR(10))

DELIMITER $$

CREATE TRIGGER st\_trig

AFTER UPDATE ON Student\_marks FOR EACH ROW

BEGIN

DECLARE aveg FLOAT;

SET aveg=(new.sub1+new.sub2+new.sub3+new.sub4+new.sub5)/5;

IF aveg>=90 THEN INSERT INTO Grades(Student\_id,average\_marks,grade)VALUES(new.student\_id,aveg,'A+');

ELSEIF aveg>=80 AND aveg<=89 THEN INSERT INTO Grades(Student\_id,average\_marks,grade)VALUES

(new.student\_id,aveg,'A');

ELSEIF aveg>=70 AND aveg<=79 THEN INSERT INTO Grades(Student\_id,average\_marks,grade)VALUES

(new.student\_id,aveg,'B+');

ELSEIF aveg>=60 AND aveg<=69 THEN INSERT INTO Grades(Student\_id,average\_marks,grade)VALUES

(new.student\_id,aveg,'B');

ELSEIF aveg>=50 AND aveg<=59 THEN INSERT INTO Grades(Student\_id,average\_marks,grade)VALUES

(new.student\_id,aveg,'C+');

ELSEIF aveg>=40 AND aveg<=49 THEN INSERT INTO Grades(Student\_id,average\_marks,grade)VALUES

(new.student\_id,aveg,'C');

ELSEIF aveg<=39 THEN INSERT INTO Grades(Student\_id,average\_marks,grade)VALUES(new.student\_id,aveg,'F');

END IF;

END $$

DELIMITER ;

UPDATE Student\_marks SET sub2=100 WHERE student\_id=4

Final examination, Spring 2019

1.b)

create database practice

create table student(std\_no int not null auto\_increment primary key,name varchar(20),dept varchar(20),year int)

insert into student (name,dept,year)values('abdul','cse',2013),('ab','eee',2015),('abc','cse',2016),

('ba','eee',2020)

create table book(isbn int not null auto\_increment primary key,title varchar(20),author varchar(20),

publisher varchar(20))

insert into book(title,author,publisher)values('ask','rahman','pask'),('bsk','jabed','pbsk'),

('csk','abdul','McGraw-Hill'),('dsk','karim','pcsk')

create table loan(stdno int,isbn int,ldate date,foreign key fk\_st(stdno)references student(std\_no),

foreign key fk\_isbn(isbn)references book(isbn))

insert into loan(stdno,isbn,ldate)values(1,3,'2019-07-01'),(3,1,'2019-08-02'),(4,2,'2019-11-30'),

(2,1,'2019-12-02'),(1,4,'2019-12-05')

update loan set isbn=3 WHERE stdno=1

1. select name from student where dept='cse'
2. select title from book where publisher='McGraw-Hill'
3. select title from book inner join loan on (book.isbn=loan.isbn)where ldate between '2019-07-01'and'2019-11-30'
4. select title,author from book inner join loan on (book.isbn=loan.isbn)inner join student on

(loan.stdno=student.std\_no) where student.name like 'a\_%\_%'GROUP BY student.name

1. select name from student inner join loan on student.std\_no=loan.stdno inner join book on

loan.isbn=book.isbn group by book.publisher having count(stdno)>5

4.b)

CREATE DATABASE practice

CREATE TABLE leap\_year(id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,YEAR INT)

INSERT INTO leap\_year(YEAR) VALUES(2001),(2010),(2000),(2100),(2024),(1500),(2300)

DELIMITER $$

CREATE FUNCTION grade(YEAR INT)

RETURNS VARCHAR(50)

DETERMINISTIC

BEGIN

if year%4=0 and year%100!=0 then return 'Leap Year';

elseif year%400=0 then return 'Leap year';

ELSE RETURN'Not leap year';

END IF;

END $$

DELIMITER ;

SELECT id,grade(YEAR) AS YEAR FROM leap\_year

6.a)

CREATE DATABASE practice

CREATE TABLE employee\_details(emp\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,fname VARCHAR(20),

lname VARCHAR(20),job\_id INT,salary FLOAT)

CREATE TABLE log\_emp\_details(emp\_id INT,salary FLOAT,edt\_date DATE)

DELIMITER $$

CREATE TRIGGER emp\_trig

AFTER INSERT ON employee\_details FOR EACH ROW

BEGIN

INSERT INTO log\_emp\_details

SET emp\_id=new.emp\_id,

salary=new.salary,

edt\_date=NOW();

END $$

DELIMITER ;

INSERT INTO employee\_details(fname,lname,job\_id,salary)VALUES('abdul','karim',1,55000.55)