Assignment 5:

Set a:

1. Object is a data structure used to store or reference data. It can be schemas, tables, aliases, views, indexes, triggers, sp, functions, global variables, etc.
2. Indexes are objects that reference table columns to provide fast data retrieval. They are used to retrieve data fast when conditions are used, sort and group data. Disadvantages include additional space, not as useful when all data tables must be retrieved and insert, update and delete are slowed.
3. Clustered and non-clustered
4. Yes, unique constraints
5. No, only one clustered index (primary key) is allowed.
6. Yes, and order matters
7. Yes
8. Normalization is organizing columns and data to make sure dependencies are enforced by integrity constraints and minimize duplicates. First, second and third normal forms
9. Adding redundant data that consequently improved performance. It can be used when many tables are used to retrieve data
10. By using error checking methods and validation procedure and suitable database design
11. Not null, check, default, unique, primary and foreign constraints
12. Primary key does not accept null values while unique key does
13. A column of data that’s a primary key in another table
14. Yes
15. It does not have to be unique and it can be null. This can happen when the value of key is not known at time of record generation
16. Indexes cant be created on table variables but can be on temporary tables
17. It’s a sequence of operations as a single logical unit of work. If it’s successful and committed, all modifications become permanent part of the database, else all modifications are lost. Transaction isolation levels are uncommitted read, committed read, repeatable read, serializable and snapshot

Set b:

Q1:

create table customer(cust\_id int, iname varchar(50))

create table order(order\_id int, cust\_id int, amount money,

order\_date smalldatetime)

select c.iname, sum(o.amount) Total

from customer c

inner join order o

on c.cust\_id = o.cust\_id

where year(order\_date) = 2002

Q2:

create table person(id int, firstname varchar(100),

lastname varchar(100))

select \* from person where lastname like 'A%'

Q3:

create table person(id int primary key, manager\_id int null,

name varchar(100) not null)

select dt.name, count(\*) from person p

left join (select \* from person p where manager\_id is null)dt

on p.person\_id = dt.manager\_id group by dt.name

Q4:

Insert, delete and update

Q5:

create table company(companyid primary key, companyname varchar(50) not null)

create table division(divisionid int primary key, companyid int foreign key, divisionname varchar(50) not null)

create table physicallocation(locationid int primary key, divisionid int foreign key,

locaddress varchar(100) not null)

create table contacts(companyid int foreign key, divisionid int foreign key,

locationid int foreign key, mail varchar(100) not null, primary key

(companyid, divisionid, locationid, mail))