Comp 3005 A1

Royce Bradley
100988820

1.

Mini-world: Some part of the real world for which the database system is developed with its data stored in the database.

Data Model: used to hide storage details and present the users with a conceptual view of the database.

Relational Data Model: all data is represented in terms of tuples (records), grouped into relations (files). Declarative operations are used to specify what to get instead of how to get them.

Database: A collection of related data stored on a computer. A generalized graph of record occurrences connected by relationships.

Database Management System: Software to facilitate the creation and maintenance of a computerized database. The purpose of a database management system is to make life easier for the user and this is achieved by hiding the complexities of the actual storage of the data from the application software.

Database System: the database, and the application programs developed on top of the DBMS. In a database system where true data independence exists it is possible to restructure the physical storage of data without invalidating any of the existing applications.

Database Schema: Description of data at some abstraction level. Each level has its own schema. There are three levels of schemas: physical, conceptual and external. Schema is also known as intension. The database schema change infrequently.

Atomic Value: values that are indivisible. A domain consists only of atomic values. All attributes of a relation are atomic.

Attribute: In a database each column has a column header that gives an indication of the meaning of the data items in that column In the formal model, the column header is called an attribute (or attribute name). Each row has an attribute (or a set of attributes) that uniquely identifies that row in the table. Each attribute has a domain or a set of valid values.

Tuple: In the formal model, rows are called tuples. A tuple is an ordered set of values (enclosed in angled brackets '< … >'). Each value is derived from an appropriate domain.

Domain: A domain has three parts: a name, a data-type/format, and a set of atomic values (indivisible). The attribute/attribute name designates the role played by a domain in a relation. It is used to interpret the meaning of the data elements corresponding to that attribute. A domain can be used for more than one attributes. Domain names and attribute names can be the same, but they have different roles in the relation. A defined type Name which is also used for an attribute also called Name.

Relation: Informally, a relation looks like a table of values. A relation typically contains a set of rows. The data elements in each row represent certain facts that correspond to a real-world entity or relationship. In the relational model all data is represented in terms of tuples (records), grouped into relations (files).

Key: Each row has an attribute (or a set of attributes) that uniquely identifies that row in the table. This is the key of the relation.

Primary key: is a chosen key. We choose a primary key by selecting the smallest of the keys (in terms of size). This is not always applicable – choice is sometimes subjective. The primary key attributes are underlined.

Foreign Key: an attribute that references the primary key of the same or different relation

DBA: Responsible for acquiring software and hardware resources, controlling its use and monitoring efficiency of operations and authorizing access to the database, for coordinating and monitoring its use.

End User: Use the database in day to day basis. Bank-tellers or airline reservation clerks do this activity for an entire shift of operations. Another example are Carleton Central users: students, instructors, etc. They don't know how the DB is structured. End-users can be categorized into: Naïve users and Business Analysts**.**

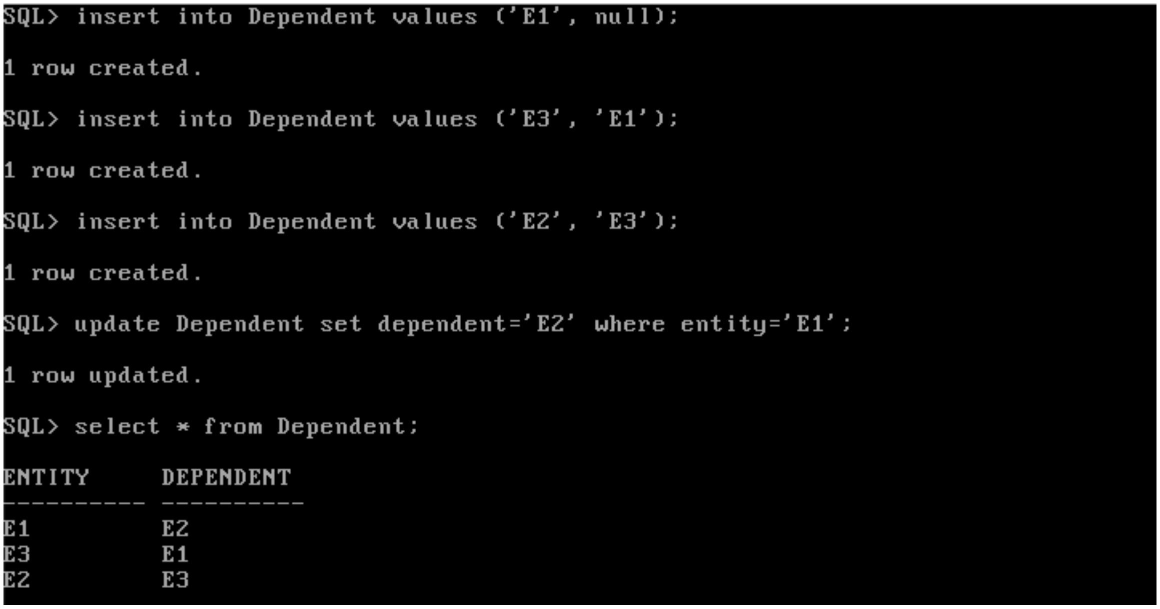
Entity Integrity Rule: The primary key attributes PK of each relation schema R in S cannot have null values in any tuple of r(R). t[PK] ≠ null for any tuple t in r(R). If PK has several attributes, null is not allowed in any of these attributes. This is because primary key values are used to identify the individual tuples.

Logical Data Independence: The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.

Query Language: Used to specify database retrievals. Includes functions such as select and clause

2.

```
Create Table Dependent(
entity          Char(10)      Primary Key,
dependent    Char(10) ,
Foreign Key(dependent) references Dependent(entity));

Insert into Dependent values(E1, null);
Insert into Dependent values(E3, E1);
Insert into Dependent values(E2, E3);

Update Dependent set dependent=E2 where entity=E1;

Select * from Dependent;
```

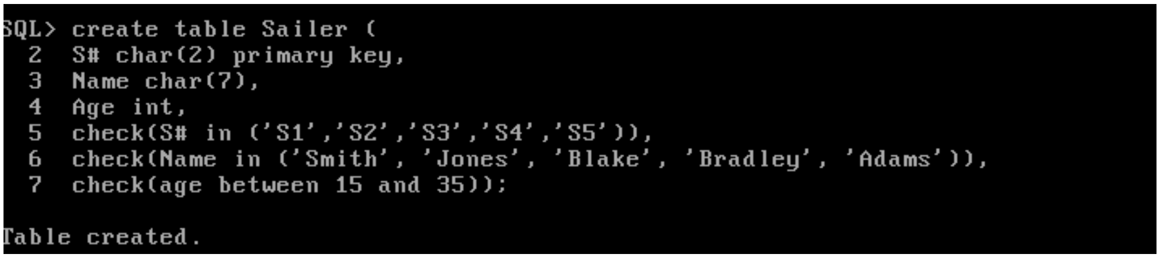


```
SQL> insert into Dependent values (’E1’, null);

1 row created.

SQL> insert into Dependent values (’E3’, ’E1’);

1 row created.

SQL> insert into Dependent values (’E2’, ’E3’);

1 row created.

SQL> update Dependent set dependent=’E2’ where entity=’E1’;

1 row updated.

SQL> select * from Dependent;

ENTITY     DEPENDENT
---------- ----------
E1         E2
E3         E1
E2         E3
```

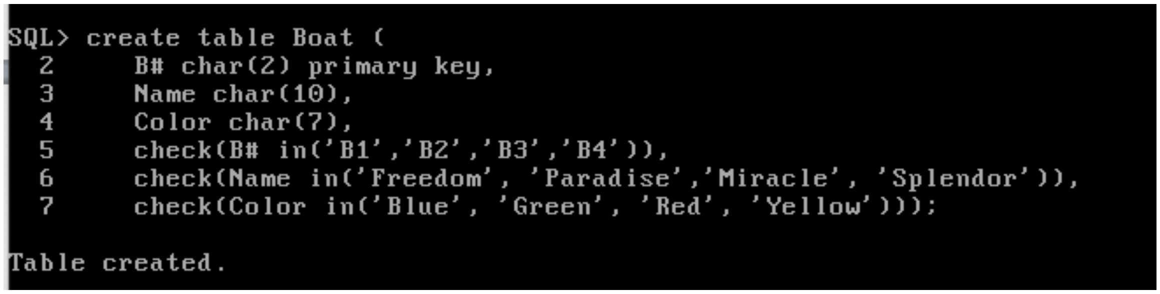
3.

```
create table  Sailer (
S# char(2) primary key,
Name char(7),
Age int,
Check(S# in (‘S1’, ‘S2’, ‘S3’, ‘S4’, ‘S5’)),
Check(Name in (‘Smith’, ‘Jones’, ‘Blake’, ‘Bradley’, ‘Adams’)),
Check(Age between 15 and 35));

create table Boat (
B# char(2) primary key,
Name char(10),
Color char(7),
Check(B# in (‘B1’, ‘B2’, ‘B3’, ‘B4’)),
Check(Name in (‘Freedom’, ‘Paradise’, ‘Miracle’, ‘Splendor’)),
Check(Color in (‘Blue’, ‘Green’, ‘Red’, ‘Yellow’)));


create table Reservation (
S# char(2),
B# char(2),
Day char(10),
Foreign key (S#) references Sailers (S#) on delete cascade,
Foreign key (B#) references Boat (B#) on delete cascade,
Check(Day in (‘1-Jan-15’, ‘2-Jan-16’, ‘3-Feb-17’, ‘4-Feb-18’, ‘5-Mar-16’, 6-Mar-17’, ‘7-
Apr-18’, ‘8-May-17’, ‘9-Jul-17’, ’10-Sep-17’)));

Insert into Sailers values (‘S1’,  ‘Smith’, 20);
Insert into Sailers values (‘S2’,  ‘Jones’, 30);
Insert into Sailers values (‘S3’,  ‘Blake’, 25);
Insert into Sailers values (‘S4’,  ‘Bradley’, 20);
Insert into Sailers values (‘S5’,  ‘Adams’, 30);

Insert into Boat values (‘B1’,  ‘Freedom’, ‘Blue’);
Insert into Boat values (‘B2’,  ‘Paradise’, ‘Green’);
Insert into Boat values (‘B3’,  ‘Miracle’, ‘Red’);
Insert into Boat values (‘B4’,  ‘Splendor’, ‘Yellow’);
Into
Insert into Reservation values (‘S1’,  ‘B1’, ‘1-Jan-15’);
Insert into Reservation values (‘S1’,  ‘B2’, ‘2-Jan-16’);
Insert into Reservation values (‘S1’,  ‘B3’, ‘3-Feb-17’);
Insert into Reservation values (‘S1’,  ‘B4’, ‘4-Feb-18’);
Insert into Reservation values (‘S2’,  ‘B1’, ‘5-Mar-16’);
Insert into Reservation values (‘S2’,  ‘B2’, ‘6-Mar-17’);
Insert into Reservation values (‘S2’,  ‘B3’, ‘7-Apr-18’);
Insert into Reservation values (‘S3’,  ‘B1’, ‘8-May-17’);
Insert into Reservation values (‘S3’,  ‘B2’, ‘9-Jul-17’);
Insert into Reservation values (‘S4’,  ‘B1’, ‘10-Sep-17’);

Select * from Reservation;
```

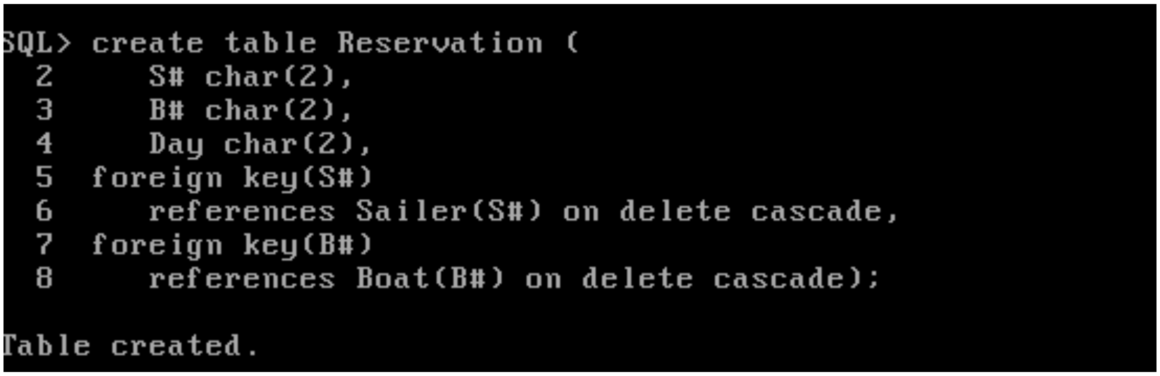
(Screen shots below)



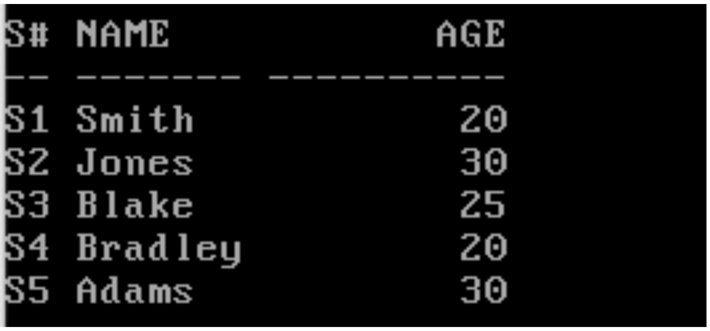
```
SQL> create table Sailer (
     S# char(2) primary key,
     Name char(7),
     Age int,
     check(S# in (’S1’,’S2’,’S3’,’S4’,’S5’)),
     check(Name in (’Smith’, ’Jones’, ’Blake’, ’Bradley’, ’Adams’)),
     check(age between 15 and 35));

Table created.
```



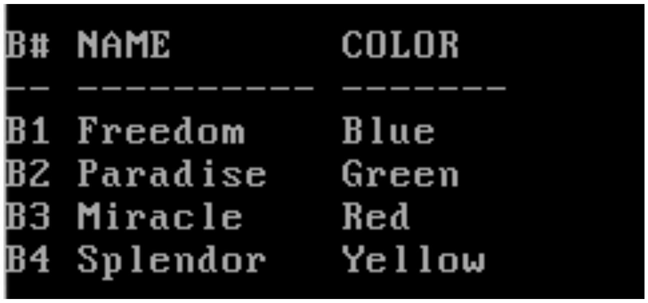
```
SQL> create table Boat (
         B# char(2) primary key,
         Name char(10),
         Color char(7),
         check(B# in(’B1’,’B2’,’B3’,’B4’)),
         check(Name in(’Freedom’, ’Paradise’,’Miracle’, ’Splendor’)),
         check(Color in(’Blue’, ’Green’, ’Red’, ’Yellow’)));

Table created.
```



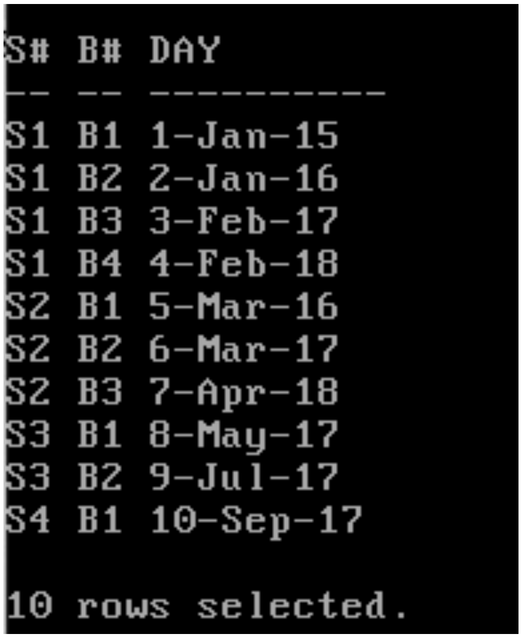
```
SQL> create table Reservation (
         S# char(2),
         B# char(2),
         Day char(2),
     foreign key(S#)
         references Sailer(S#) on delete cascade,
     foreign key(B#)
         references Boat(B#) on delete cascade);

Table created.
```



```
S# NAME           AGE
-- ------- ----------
S1 Smith           20
S2 Jones           30
S3 Blake           25
S4 Bradley         20
S5 Adams           30
```



```
B# NAME       COLOR
-- ---------- -------
B1 Freedom    Blue
B2 Paradise   Green
B3 Miracle    Red
B4 Splendor   Yellow
```



```
S# B# DAY
-- -- ----------
S1 B1 1-Jan-15
S1 B2 2-Jan-16
S1 B3 3-Feb-17
S1 B4 4-Feb-18
S2 B1 5-Mar-16
S2 B2 6-Mar-17
S2 B3 7-Apr-18
S3 B1 8-May-17
S3 B2 9-Jul-17
S4 B1 10-Sep-17

10 rows selected.
```