COMP 3007 Assignment 1

Sam Diamantstein 101060342

COMP 3007, Sept 21, 2017

Part 1 Concepts

1. Data

Are known facts that can be recorded and that have an implicit meaning

2. Mini World

Some part of the real world for which a database system is developed to represent it. Changes to the mini world are reflected in the database

3. Database System

Refers to the database, and the application programs developed on top of the DBMS.

DBMS

The collection of programs used to create maintain, and provide controlled access to a database.

5. DBA

Database Administrators are responsible to acquire the software and hardware resources needed to create the database, to control the databases use and monitor the efficient of operations, authorize access to the database and coordinate and monitor its use. The DBA is accountable for problems like security breaches and poor response times.

6. End User

The individual who use the database day to day, who do not know how it is constructed, and interact with it by using an application, either as a naive user or as a business analyst.

7. Business Analyst

Business analysts are people who analyze business data and large data sets to improve decision making for business operations. They do so by using tools that can manipulate and query the database they are interested in.

8. Data Model

It is a type of data abstraction that captures the nature and relationships among data into a conceptual representation. The data model uses logical concepts, such as objects, their properties, and their interrelationships, that may be easier for most users to understand than computer storage concepts. The data model hides storage and implementation details that are not of interest to most database users.

9. Relational Data Model

is an approach to managing data using a structure and language consistent with first-order predicate logic, first described in 1969 by Edgar F. Codd. It organizes

data is the form of tables and relationships among the data stored in them. here all data is represented in terms of tuples, grouped into relations. A database organized in terms of the relational model is a relational database.

10. Atomic Value

The value of an attribute that is indivisible, and is also called a simple value.

11. Domain

It is the value set that each simple attribute of an entity type is associated with. It species the set of values that may be assigned to that attribute for each individual entity.

12. Key

A key is used to establish and identify relation between tables. It also ensure that each record within a table can be uniquely identified by combination of one or more fields within a table.

13. Foreign Key

A foreign key is a key used to link two tables together and is a field (or collection of fields) in one table that refers to the primary key in another table.

14.Relation

It is a named two-dimensional table of data. Each relation (or table) consists of a set of named columns and an arbitrary number of unnamed rows.The instance of a relation is a set (unordered, atomic values, with null values for unknowns or not applicable). This represents how data is related in the mini world.

15. Attribute

represents some property of interest that further describes an entity (like a name, or student number). More formally, this is a map from the relation name to the domain of possible values.

16. Tuple

is the proper name for a table row in the formal relational model termi-nology. More formally a tuple is an ordered set of values (enclosed in angled brackets ¡ ... ¿) where value is derived from the appropriate domain.

17. Entity Integrity Rule

It ensures that every relation has a primary key and that the data values for that primary key are all valid. In particular, it guarantees that every primary key attribute is non-null.7

18. Logical Data Independence

It is the capacity to change the conceptual schema without having to change external schemas or application programs.

19. Data Denition Language (DDL)

it is used by the database designers and DBA to specify the conceptual schema

of a database. In many DBMSs, the DDL is also used to dene internal and external schemas (views).

20. Data Manipulation Language (DML)

A data manipulation language (DML) is a family of syntax elements similar to a computer programming language used for selecting, inserting, deleting and updating tuples in a database.

Please find next two parts below

– – PART 2.

```
 – – create the tables

CREATE TABLE qOne
    (
    Object char(255) NOT NULL,
    Dependent char(255) NOT NULL,
     CONSTRAINT obj_pk PRIMARY KEY (Object),
     CONSTRAINT dep_fk FOREIGN KEY (Dependent)
    REFERENCES qOne(Object) DISABLE
);

– – insert content into the table

   INSERT ALL
    INTO qOne(Object, Dependent) Values ('e2','e1')
    INTO qOne(Object, Dependent) Values ('e3','e2')
    INTO qOne(Object, Dependent) Values ('e1','e3')
    SELECT * FROM dual;

 – – modify the table, enable the constraint
ALTER TABLE qOne
ENABLE CONSTRAINT dep_fk;
```

– – PART 3.

```
                – – CREATE THE THREE TABLES

SQL> CREATE TABLE Suppliers (
  2  S# varchar(255) NOT NULL,
  3  SNAME varchar(255),
  4  STATUS int,
  5  CITY varchar(255),
  6  PRIMARY KEY (S#)
  7  );


SQL> CREATE TABLE Parts(
  2  P# varchar(255) NOT NULL,
  3  PNAME varchar(255),
  4  Color varchar(255),
  5  Weight int,
```

```
  6  City varchar(255),
  7  PRIMARY KEY (P#)
  8  );

SQL> CREATE TABLE sp (
  2  S# varchar(255),
  3  p# varchar(255),
  4  QTY int,
  5  FOREIGN KEY ( S#) references Suppliers (S#),
  6  FOREIGN KEY ( p#) references Parts (P#),
  7  PRIMARY KEY (S#)
  8  );
```

– – SET ADDITIONAL CONSTRAINTS

```
– – check that the status of the supplier is greater than zero
ALTER TABLE Suppliers
ADD CONSTRAINT check_Status (status > 0);


– – ensure that the name of the supplier is not null
ALTER TABLE Suppliers
MODIFY SNAME varchar NOT NULL;

– – check that the only colours to be used for the parts  tables are
red, blue or green

ALTER TABLE Parts
ADD CONSTRAINT check_color CHECK (Color IN ('Red', 'Blue', 'Green'));


 – – check that the #p value has two character
ALTER TABLE Parts
ADD CONSTRAINT check_hashP CHECK (LENGTH(P#) = 2);

– – check that the qty for the supplier part is at least 100 to 400
units
ALTER TABLE sp
ADD CONSTRAINT check_qty CHECK (QTY BETWEEN 100 and 400);
```

– – POPULATE TABLES

```
INSERT ALL
```

```
INTO Suppliers (s#, sname, status, city) VALUES ('s1',
'Smith', 20, 'London')
INTO Suppliers (s#, sname, status, city) VALUES ('s2',
'Jones', 30, 'Paris')
INTO Suppliers (s#, sname, status, city) VALUES ('s3',
'Blake', 30, 'Paris')
INTO Suppliers (s#, sname, status, city) VALUES ('s4',
'Clark', 20, 'London')
INTO Suppliers (s#, sname, status, city) VALUES ('s5',
'Adams', 30, 'Athens')

INTO Parts (p#, pname, color, weight, city) VALUES ('p1',
'Nut', 'Red', 12.0, 'London')
INTO Parts (p#, pname, color, weight, city) VALUES ('p2',
'Bolt', 'Green', 17.0, 'Paris')
INTO Parts (p#, pname, color, weight, city) VALUES ('p3',
'Screw', 'Blue', 17.0, 'Oslo')
INTO Parts (p#, pname, color, weight, city) VALUES ('p4',
'Screw', 'Red', 14.0, 'London')
INTO Parts (p#, pname, color, weight, city) VALUES ('p5',
'Cam', 'Blue', 12.0, 'Paris')
INTO Parts (p#, pname, color, weight, city) VALUES ('p6',
'Cog', 'Red', 19.0, 'London')

INTO sp (s#, p#, qty) VALUES ('s1', 'p1', 300)
INTO sp (s#, p#, qty) VALUES ('s1', 'p2', 200)
INTO sp (s#, p#, qty) VALUES ('s1', 'p3', 400)
INTO sp (s#, p#, qty) VALUES ('s1', 'p4', 200)
INTO sp (s#, p#, qty) VALUES ('s1', 'p5', 100)
INTO sp (s#, p#, qty) VALUES ('s1', 'p6', 100)
INTO sp (s#, p#, qty) VALUES ('s2', 'p1', 300)
INTO sp (s#, p#, qty) VALUES ('s2', 'p2', 400)
INTO sp (s#, p#, qty) VALUES ('s3', 'p2', 200)
INTO sp (s#, p#, qty) VALUES ('s4', 'p2', 200)
INTO sp (s#, p#, qty) VALUES ('s4', 'p4', 300)
INTO sp (s#, p#, qty) VALUES ('s4', 'p5', 400)

 SELECT * FROM dual;


- - total constraints = Primary Key + Foreign Key + NOT
NULL + Check_Status + Modify SName + Check_color +
check_hashP + check_qty = 8 unique constrains
```