Andrew Mackay

ID:3441988

Introduction to database management

Assignment 1

1. a) Data independence is when you can modify the scheme or database without affecting the program/applications ability to function. Data is made separate from the program as to ensure that any changes made to it will not affect the programs ability to perform, this can be thought of as “abstraction”. The two types of data independence are physical data independence and logical data independence. In physical data independence the logical and conceptual levels are not affected by physical changes such as A new storage device, new data structure, changing database locations, and a different different data access method. Logical data independence is another way of saying the users view of data. Changes in the logical level are a change in the data definition, adding, deleting and updating new attributes entities or relationships in the database.(tutorialspoint, 2022)

b) Software development life cycle, like the name may imply, is a software development approach where the steps to the development are clearly defined in stages and generally this is a low cost technique. “Development Prototype model vs SDLC requires the development team to make an extra effort. Software prototyping is used to help users evaluate their developer proposals and test them before implementation.”(isetech, 2022) The SLDC is considered the traditional methodology to developing, maintaining and replacing information systems. The specific steps of SDLC are planning, analysis, design, implementation, and maintenance. The SDLC can only commence being completed after the organizations findings from their enterprise data modeling.

c) An entity is a real world object as opposed to the relationship which is an association between entities.(pediaa, 2018) An entity can be many different things like a person, place, thing or concept. There exist established relationships between entities which form the relationships and relational databases, and at the core of these are the entries of data which are categorized as entities.

d) Three schema architecture aims to give data access to separate groups of people who each have their own unique view of that data depending on which level they are in. Three-Schema architecture breaks relational databases into 3 different categories. The external schema, conceptual schema, and internal schema. (javatpoint, 2022) This system can be implemented within the SDLC for further clarity, which is especially useful when considering the broader range of stakeholders.

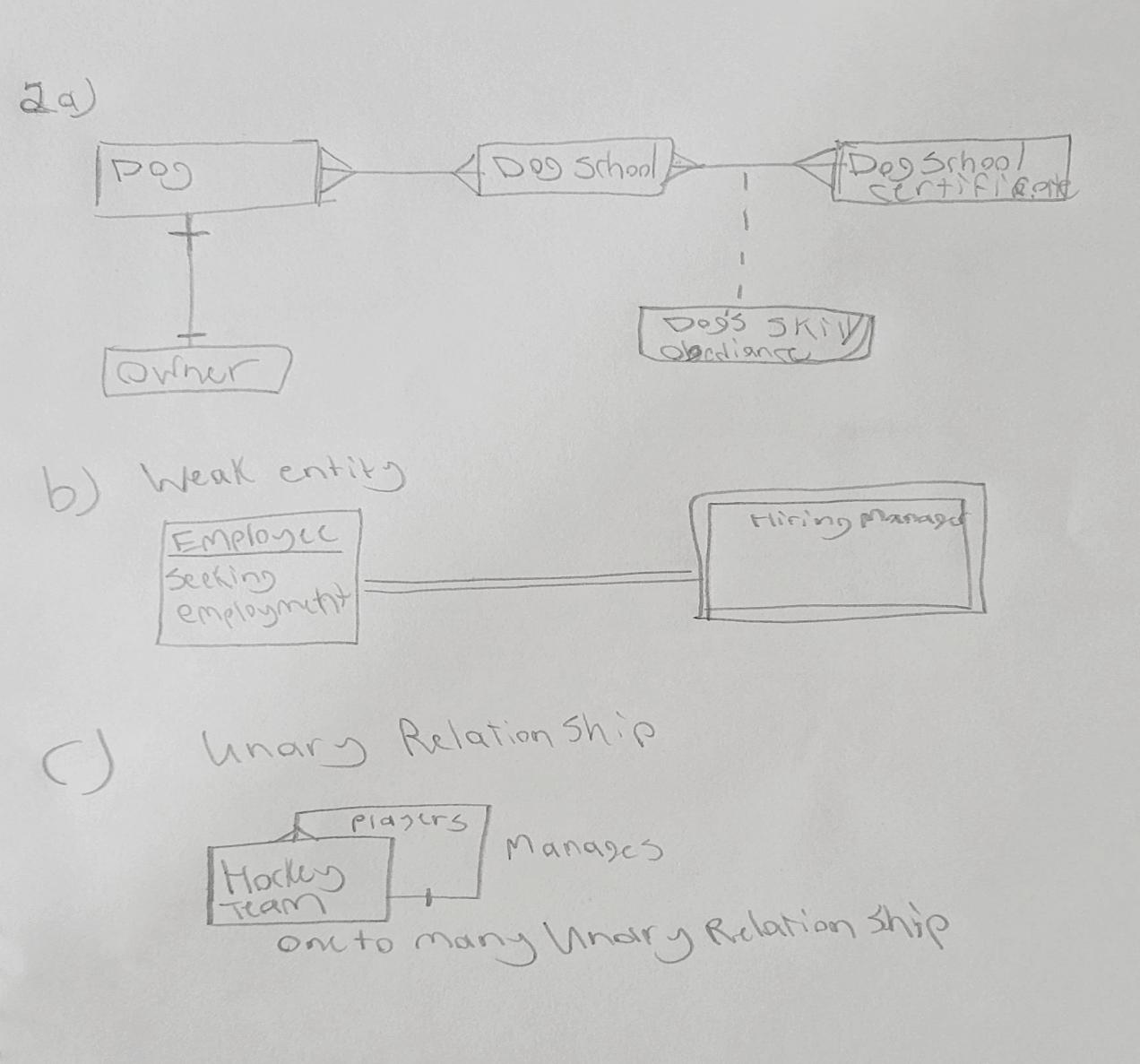
The External schema refers to the views of employees and managers who use the database in which this view is comprised of enterprise data or otherwise known as top down view, and a collection of detailed user views otherwise known as bottom up. The Conceptual Schema represents the view which the data architect or data administrator has, and combines all external views into a single combined definition of the enterprises data.

The internal Schema is a combination of logical and physical schema. Logical being the representation of data, while the physical schema shows how the data is to be represented and stored.

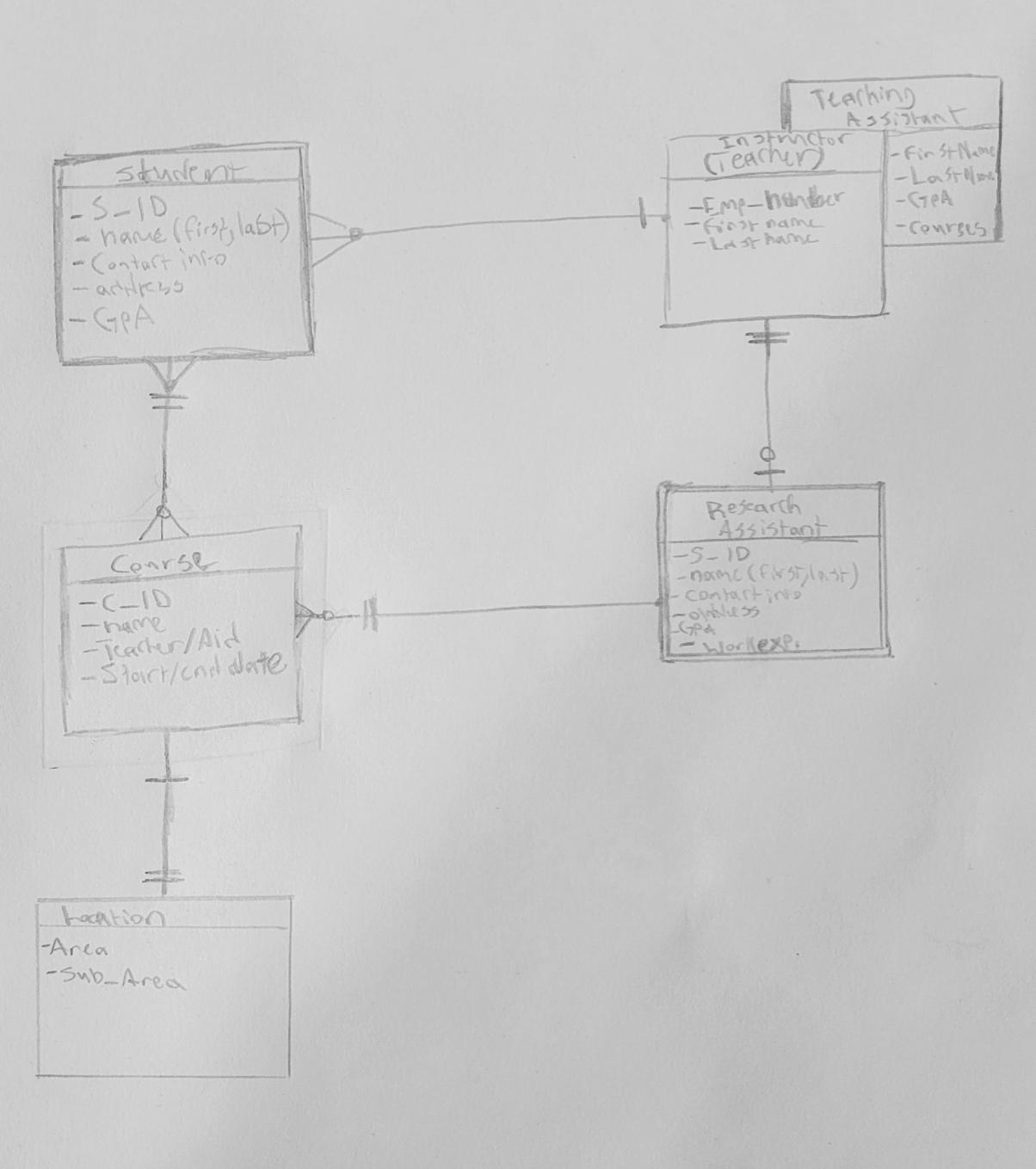
e) A database stores data with the capabilities of searching and reporting on structured data from one source. (confluent, 2022) Reporting, retrieving and updating data can be done easily with SQL Querying within a database.

In a data warehouse, a vast amount of structured data is stored. This data storage types content comes from a variety of operational databases.

A data lake is a conglomeration of various types of data and has no restrictions or organizations of data structures. The data lake is an integrated repository for external and internal data which also doesn’t follow any predefined schema.



3.



4.

Table:Vendor

CREATE TABLE Vendor(

Vendor\_ID int,

Vendor\_Name String

Vendor\_Address number

Primary key(Vendor\_ID));

Table:Project

CREATE TABLE Project(

Project\_No int,

Project\_Location int,

constraint Project\_pk Primary key(Project\_No),

constraint Project\_Location\_Fk foreign key(Project\_Location) references Location(Loc\_ID));

Table: Skill

create table Skill(

Skill\_No int,

Skill\_Description String,

Primary key(Skill\_No));

Table: Location

CREATE TABLE Location(

Loc\_ID int,

Loc\_City string,

Loc\_State string,

Loc\_Population int,

Primary key(Loc\_ID));

Table:Department

CREATE TABLE Department(

Dept\_No int,

Dept\_Name varchar(128),

Dept\_Address varchar(128),

Primary key(Dept\_No));

Table: Employee

CREATE TABLE Employee(

Emp\_No int,

Emp\_Name String,

Title String,

Date\_Of\_Birth date,

is\_married\_to int,

date\_married date,

Dept\_No int,

Project\_No int,

Primary key(Emp\_No),

constraint Emp\_no foreign key(inner join) references Employee(Emp\_No),

constraint Employee\_Project\_FK foreign key(Project\_No) references Project(Project\_No));

5.

1. “Referential integrity requires that a foreign key must have a matching primary key or it must be null. It means the reference from a row in one table to another table must be valid.”(opentext, 2014)

The E\_id column is shared between the Employee and Works tables whereas dept ID is in place of Emp ID within the parent Department row, but the referential relationship does not depend on similar names rather their position in the table and established dependence from child - parent.

Referential integrity exists in this instance where each column is matched by a key within the next corresponding one where there is established dependence. Each table from child - child - parent, reference one another using the Id key associated with each section and line up with the corresponding referential section

B)

“In a relational database, a tuple is one record (one row).”(pcmag, 2022)

There are generally 4 options when deleting a tuple and in this case the “dept” tuple specifically. These options are:

To allow the deletion of the dept tuple if a some of a tables data refer to it, to set the ID fields that refer to it to null, delete all tuples that refer to the dept tuple, set the referencing ID fields to a default pre existing dept.

1. A)

In 1NF or first normal form, there aren’t repeating values, they are considered atomic or singular also. There is also a defined primary key.

Within 2NF, or second normal form, this is defined by a relation being otherwise in 1NF but with fully functional, dependencies between the nonkey attributes to the primary key. Fully functional dependencies are also known as no partial dependencies.

The 3NF third normal form, is when the relationship is otherwise in second normal form but with the additional feature of no transitive dependencies. A transitive dependency is a non key link between and one or more non key attributes in which depend on the primary key.

So in viewing dependencies we take logical steps to determine where the place of which thing is based on its dependencies spot in the table. The things that all others depend on tend to be at the furthest left at the start of the table, as we read from left to right. Then going in order of importance in almost like a ladder of dependencies from left to right we can determine the place of the other columns. To go from 1NF,2NF,3NF we are going to be eliminating columns and displacing them within separate tables.

STUDENT(StuID, StuName, Address, Birth\_Date, Gender, Major)

All of the values are atomic therefor the table is already in 1NF, StuID is primary key, no duplicated rows or columns.

To make this into 2NF we can split some of the academic and student info details by dependency/category

STUDENT(StuID, StuName, Major)

STUDENTINFO( StuID, StuName, Gender, Address, Birth\_Date )

This also meets the 3NF criteria as there are no transitive dependencies.

B)

Consider the relation EMPLOYEE, where an employee can have more than one specialization and more than one dependent. EmpID is a unique identifier for employees and dependent\_name is a unique identifier for each employee’s dependents; however, dependents of different employees may have the same name

1NF

RELATION = EMPLOYEE (EmpID, Name, Phone, Email, Address, Specialization, Dependent\_name, Dependent\_age, Dependent\_gender)

2NF/3NF

EMPLOYEE((EmpID, Name, Phone, Email, Address, Specialization)

EMPLOYEEDEPENDANT(Dependent\_name, Dependent\_age, Dependent\_gender)

There are atomic values for all attributes making this relation in the 1NF format.

To then convert from 1NF to 3NF, a jump that is the most common conversion to make however still needs to be converted to 2NF before going to 3NF. This can be done in a few steps.

Non key attributes should have fully functional dependence in relation to the emp\_no which is the primary key to convert from 1NF - 2NF. Then for 3NF we remove the transitive dependencies.

All business rules in place for this question assure the atomic nature of the original table values.

(emp\_no, date, job,phone\_no, office\_no, area, sal,proj\_no, p\_budget, dep\_no, d\_budget, mgr\_emp\_no).

This is the original relation and if we are to split it into 2NF it would look more like this:

EMPLOYEE(emp\_no,phone\_no,office\_no,dep\_no,proj\_no)

JOB(emp\_no,date,job,sal)

In 3NF we can reintroduce columns that were stripped previously, while removing the transitive dependencies. By keeping those previous 2 tables otherwise the same we can split the transitive dependencies which are the office and office phone into their own tables as

OFFICECONTACT(phone\_no, office\_no)

OFFICE(office\_no, area, dep\_no)

“No office can be assigned to more than one department at a time, phone\_no → office\_no, office\_no → area, office\_no → dep\_no”.

PROJECT(proj\_no, dep\_no, budget)

“proj\_no → dep\_no, proj\_no → p\_budget”

DEPARTMENT(dep\_no,mgr\_emp\_no, d\_budget)

DEPARTMENTMANAGER(mgr\_emp\_no, dep\_no)

“dep\_no → mgr\_emp\_no, dep\_no → d\_budget”

References

Hoffer, J., Venkataraman, R., & Topi, H. (2018). Modern Database Management (13th ed.). Pearson Education (US). https://online.vitalsource.com/books/9780134792293

What is Data Independence in DBMS?. (2022). Retrieved 26 October 2022, from <https://www.tutorialspoint.com/what-is-data-independence-in-dbms>

SDLC vs Prototyping – The Ultimate Comparison. (2022). Retrieved 26 October 2022, from <https://isetech.co/sdlc-vs-prototyping/>

Difference Between Entity and Relationship in DBMS - Pediaa.Com. (2018). Retrieved 26 October 2022, from <https://pediaa.com/difference-between-entity-and-relationship-in-dbms/#:~:text=The%20main%20difference%20between%20entity,or%20diamond%20represents%20a%20relationship.>

DBMS Three schema Architecture - javatpoint. (2022). Retrieved 27 October 2022, from <https://www.javatpoint.com/dbms-three-schema-architecture>

Databases, Data Lakes, and Data Warehouses Explained. (2022). Retrieved 27 October 2022, from <https://www.confluent.io/learn/databases-data-lakes-and-data-warehouses-compared/>

Eng, A. (2014). Chapter 9 Integrity Rules and Constraints. Bccampus. Retrieved from <https://opentextbc.ca/dbdesign01/chapter/chapter-9-integrity-rules-and-constraints/>

Definition of tuple. (2022). Retrieved 5 November 2022, from <https://www.pcmag.com/encyclopedia/term/tuple>

Database Normalization – Normal Forms 1nf 2nf 3nf Table Examples. (2022). Retrieved 18th December 2022, from https://www.freecodecamp.org/news/database-normalization-1nf-2nf-3nf-table-examples/amp/