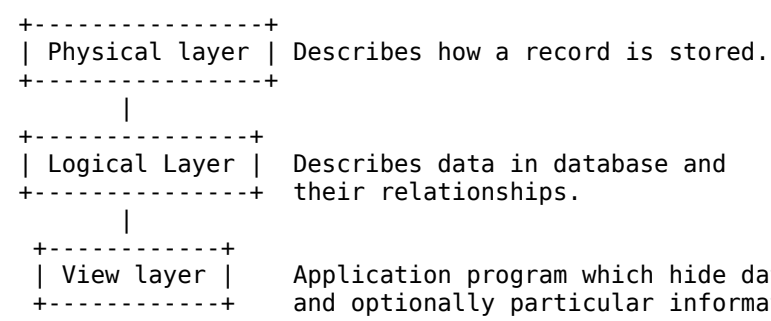
Databases and ER-diagrams
Sander van Veen (6167969)

Question 1:



Question 2:

DBMS
DBMS is short for database management system and is the general term for a system used to manipulate a database.

DBA
DBA is an abbreviation for Database administrator, also known as a supervisor. His job is to create, manage and tweak the database design and some of its implementation (for example, the way the data is stored).

DML
Data Manipulation Languages are used to manipulate a database. SQL (structured query language) is most widely used. There are two classes of DML's:
- Procedural, where a user specifies what data is required, and how it should be found.
- Declarative, where a user specifies what data is required, but does not specify how to find the data.

DDL
A Data Definition Language can describe how the data will be stored. It uses metadata (in a schema) to separate the records.

Data model
Collection of tool describing data, relationships, semantics and constrains (rules used to ensure data integrity).

Normalization
Separating repeated data into different tables, to prevent redundancy and bad database design.

Query processing
User specifies a Query, which will start the following sequence:
=> Parser / Translator (convert SQL to algebra)
=> Relational algebra expression
=> Optimizer (with statistics about data)
=> Execution plan (how to find the data)
=> Evaluation engine (seeks through data)
=> Query output (returns the results)

Query optimization
Convert queries to better ones, to improve performance. For example, only select records from tables which do matter, before performing a Join-operation, instead of using all records.

_____ Question 3: _____

The design process consists of the following phases:

- The initial phase gives a designer a better feeling about the data, which will be used and is called "Specification of user requirements". For example, this is the moment where you ask the manager of a supermarket about what data is used and what the database should be able to do.

- The second phase is "conceptual-design", where the user requirements will be translated to a conceptual database schema. This schema is a detailed overview of the enterprise and used to confirm that all user requirements are met. The relationships between the data is also described in this phase.

  There are two ways to visualize the grouping of the various attributes, used in the database schema:

  - An entity relationship diagram, where an entity is an object (rectangle) and various shapes (diamond, ellipse, or double bordered) and (un)directional lines represent the relationships between the entities.

  - Normalization, which is a set of algorithms used to generate tables.

- When the conceptual schema is designed, you can define a "specification of functional requirements". This specification describes what kind of operations the users can and cannot perform. For example, an employee of a supermarket can (in most cases) not view the salary of the other employees. But a manager has (in some cases) the authority to manipulate the salary of other employees.

- Once you have setup the functional requirements, you can start implementing the design of the database. This phase is called "Logical design phase" and differs from the "Conceptual design", because in this phase you can, for example, specify the data types used for storing the data.

- The last phase is the "Physical design", where the features of the database are specified. This phase also specifies the database engine (e.g. InnoDB, MyISAM) to use and the internal storage structures (for example, file organisation).

_____ Question 4: _____

4.1. Entity sets

The six rectangles (department, project, location, employee, manager and family) are entities. Entities can be thought of as nouns. Every entity (except for weak entities) have a primary key, which can be used for identification. An entity set is a collection of multiple instances of an entity.

4.3. Relationship sets

In order to say something about two entities, they need a relationship. A collection of multiple relationships between the same entities is called a relationship set. In the ER diagram, these relationship sets are shown as grey diamonds and they connect the entities using two or more lines (depending on the relationship).

4.5. Attributes for entity sets

A simple attribute is one component that is atomic. For example, project-number is a property of a Project. Attributes are shown as ellipses in a ER diagram.

4.6. Primary keys

A primary key used to identify a member of an entity set. In a ER diagram, an attribute with an underlined name is (part of) a primary key.

4.7. Composite attributes

A composite attribute has multiple components, each of which is atomic or composite.
For example, an attribute "name" is a composition of first- , given- and family name.
Composite attributes are shown as attributes of attributes, thus two or more ellipses,
where each is connected using a line to a single ellipse.

There are no composite attributes visible in the ER diagram of this assignment, because
they would make the diagram more difficult to read and thus irrelevant. An other example
of a composite attribute is "address", which can be split into street, city and house number.

4.8. Multi-valued attributes

A multi-valued attribute has more than one value for a particular entity. There are no
multi-valued attributes specified in the assignment and therefore not visible in the ER
diagram. However, a multi-valued attribute is shown as a ellipse with two borders.

4.9. Derived attributes

A derived attribute can be obtained from other attributes or related entities. The most
common example is the age of a person. It would be ridiculous to update the database for
each user, when a user celebrates his birthday. The age of a person can be obtained from
the user's date of birth. A derived attribute is shown as a dotted ellipse and line.

4.13. Weak entity sets

A weak entity is an entity, which relays on another entity. It is weak, because it cannot
be an entity on its own. Therefore, entities which are on its own, are called strong.

Weak entities are shown as normal entities, but with a second border.

4.14. Weak relationship sets

Weak relationships are relationships, which connect at least one weak entity to one or more
entities. Just like weak entities, weak relationships are shown as a normal relationship,
but with a second border. Please note that a weak relationship connected to a strong entity,
has a normal line and a double line when connected to a weak entity. Thus, the type of line
(double line or single line) refers to the connected type of entity (weak or strong).