Inhoud

[Database Hoofdstuk 1 – Introduction 3](#_Toc474933341)

[Hoofdstuk 1.1 Introduction 3](#_Toc474933342)

[Course goals 3](#_Toc474933343)

[Hoofdstuk 1.2 Introduction to Databases 3](#_Toc474933344)

[Data Versus information 3](#_Toc474933345)

[Database Definition 3](#_Toc474933346)

[Database Management System 3](#_Toc474933347)

[Key computing Terms 3](#_Toc474933348)

[Transformation in computing 4](#_Toc474933349)

[History of the Database Timeline 4](#_Toc474933350)

[Hoofdstuk 1.3 Types of Database Models 5](#_Toc474933351)

[Database Development Process 5](#_Toc474933352)

[Common Types of Database Models 5](#_Toc474933353)

[Flat File Model 5](#_Toc474933354)

[Hierarchical Model 6](#_Toc474933355)

[Network Model 6](#_Toc474933356)

[Relational Model 7](#_Toc474933357)

[Object-Oriented Model 7](#_Toc474933358)

[Hoofdstuk 1.4 Relational Databases and Normalization 8](#_Toc474933359)

[Introduction to Relational Databases 8](#_Toc474933360)

[Rules for Relational Databases 8](#_Toc474933361)

[Normalization 8](#_Toc474933362)

[Objectives of Normalization 8](#_Toc474933363)

[Advantages of a Relational Database 9](#_Toc474933364)

[Data integrity 9](#_Toc474933365)

[Hoofdstuk 1.5 Database Storage Structures 9](#_Toc474933366)

[Database Data Storage 9](#_Toc474933367)

[Introduction to logical structures 9](#_Toc474933368)

[Introduction to Physical Storage Structures 10](#_Toc474933369)

[Hoofdstuk 1.6 Business Requirements 11](#_Toc474933370)

[Why do I need a Database Solution? 11](#_Toc474933371)

[Importance of Business Rules 11](#_Toc474933372)

[Database Hoofdstuk 2 – Databases and Data Modeling 11](#_Toc474933373)

[Hoofdstuk 2.1 What is a Table? 11](#_Toc474933374)

[Single table 11](#_Toc474933375)

[Advantages and disadvantages of a flat file database 11](#_Toc474933376)

[Hoofdstuk 2.2 Relational Databases 12](#_Toc474933377)

[Relational Databases 12](#_Toc474933378)

[Advantages 12](#_Toc474933379)

[Key terms 12](#_Toc474933380)

[Properties of tables 12](#_Toc474933381)

[Hoofdstuk 2.3 Conceptual Data Modeling 12](#_Toc474933382)

[conceptual modeling 12](#_Toc474933383)

[Entities: 13](#_Toc474933384)

[Creating 13](#_Toc474933385)

[Entity Relationship model 13](#_Toc474933386)

[Goals 13](#_Toc474933387)

[Creating 13](#_Toc474933388)

[Hoofdstuk 2.4 Data Modeling Terminology 13](#_Toc474933389)

[Table diagram Notations 13](#_Toc474933390)

[Naming conventions for tables 13](#_Toc474933391)

[Naming conventions for Columns 13](#_Toc474933392)

[Table Short Names 13](#_Toc474933393)

[Naming Restrictions with Oracle 14](#_Toc474933394)

[Hoofdstuk 2.5 Unique identifiers and Primary Keys 14](#_Toc474933395)

[Conceptual Data model UID 14](#_Toc474933396)

[Primary key (PK) 14](#_Toc474933397)

[Composite UID and Primary Key 14](#_Toc474933398)

[Hoofdstuk 2.6 Relationships and Foreign Keys 14](#_Toc474933399)

[Relationships 14](#_Toc474933400)

[Foreign key (FK) 15](#_Toc474933401)

[Hoofdstuk 3 – Creating a Physical Data Model 15](#_Toc474933402)

[Hoofdstuk 3.1 Conceptual and Physical Data Models 15](#_Toc474933403)

[What is a Conceptual Model? 15](#_Toc474933404)

[What is a Logical Model? 15](#_Toc474933405)

[What is a Physical Model? 15](#_Toc474933406)

[Steps to create a physical data model 15](#_Toc474933407)

[Conceptual and Physical Models 15](#_Toc474933408)

[Hoofdstuk 3.2 Business Rules 15](#_Toc474933409)

[Business Rules 15](#_Toc474933410)

[Source of Business Rules 16](#_Toc474933411)

[Structural Business Rules 16](#_Toc474933412)

[Procedural Business Rules 16](#_Toc474933413)

[Additional Programming 16](#_Toc474933414)

# Database Hoofdstuk 1 – Introduction

## Hoofdstuk 1.1 Introduction

### Course goals

1. Describe the purpose of a relational database
2. Desribe key business requirements when developing a database
3. Use data modeling to architect a relational database
4. Develop an entity relationship diagram (ERD) – model data
5. Use Oracle SQL Developer Data Modeler to build ERDs
6. Make a physical model from an ERD
7. Create a physical model from a logical model (ERD)
8. Write, execute, and save SQL statements in Oracle Application Express

## Hoofdstuk 1.2 Introduction to Databases

### Data Versus information

Data is collected facts about a topic or item

Information is the result of combining, comparing, and performing calculations on data

### Database Definition

1. Is a centralized and structured set of data stored on a computer system
2. Provides facilities for retrieving, adding, modifying, and deleting the data when required
3. Provides facilities for transforming retrieved data into useful information

### Database Management System

DBMS is a software system that controls the storage, organization, and retrieval of data

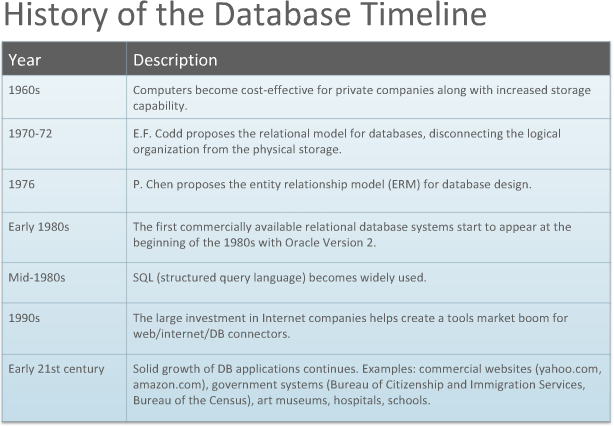
### Key computing Terms

1. Hardware
2. Software
3. Operating system
4. Application
5. Client
6. Server

### Transformation in computing

1. 1970s: Mainframe computing
   1. A mainframe(Database) connects “Dumb Terminals”
2. 1980s: Desktop Computing
   1. Software(DataBase) connects “smart clients: GUI Interface and Software””
3. 1990s: Client/Server Computing
   1. Software sever(database) connect application server(software) that connects “Thin Clients: GUI interface, Browers”
4. 2000s>: Grid Computing
   1. A Database and Web Users go to “news, sports, entertainment, movie, games”” etc

### History of the Database Timeline



## Hoofdstuk 1.3 Types of Database Models

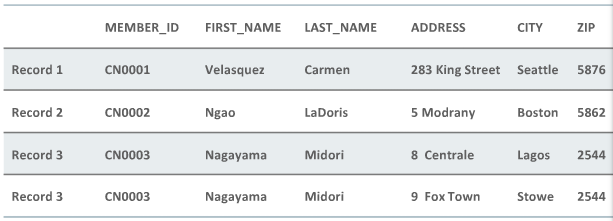
### Database Development Process

Conceptual Data Modeling 🡪 Database Design 🡪 Databas Build 🡪 Database

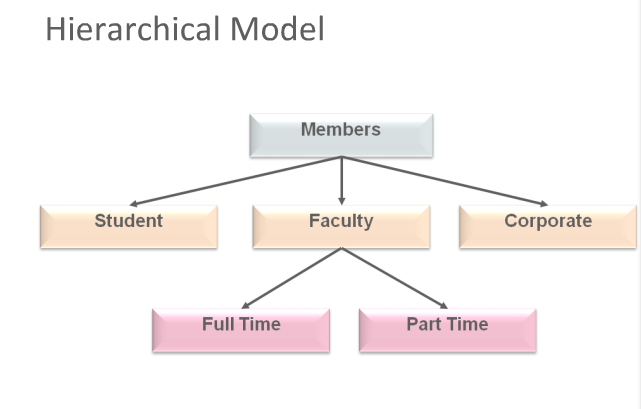
### Common Types of Database Models

1. Flat File Model
2. Hierarchical Model
3. Relational Model
4. Network Model
5. Object-Oriented Model

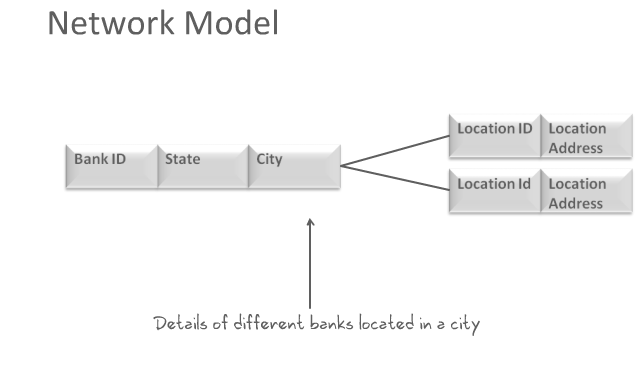
### Flat File Model



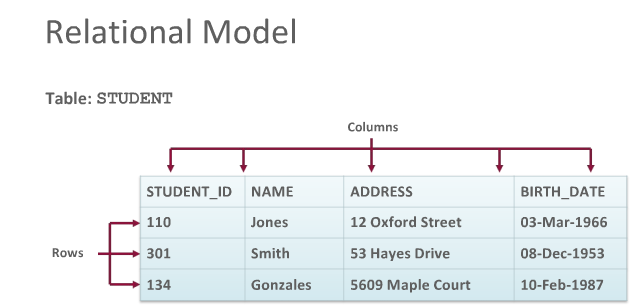
### Hierarchical Model



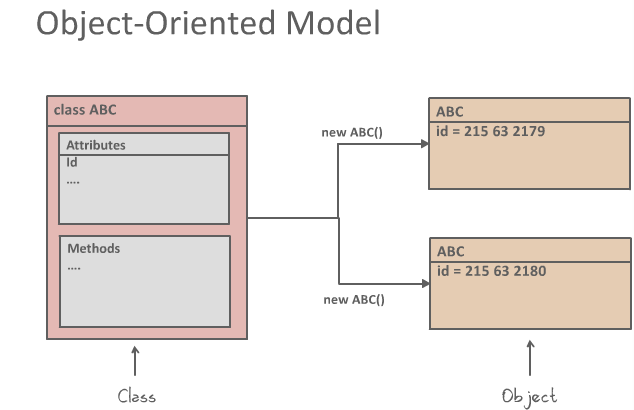
### Network Model



### Relational Model



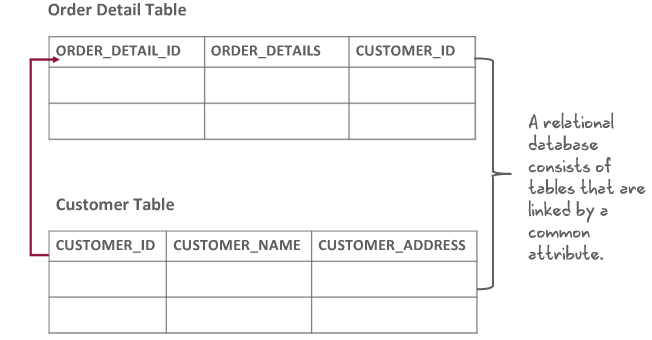
### Object-Oriented Model



## Hoofdstuk 1.4 Relational Databases and Normalization

### Introduction to Relational Databases

1. A relational database stores information in tables with rows and columns
2. A table is a collection of records
3. A row is called a record (or instance)
4. A columns is referred to as a field(or attribute)



### Rules for Relational Databases

1. Each table has a distinct name
2. Each table may contain multiple rows
3. Each table has a value to uniquely identify the rows
4. Each column in a table has a unique name

### Normalization

1. Is the process of organizing the attributes and tables of a relational database to minimize redundancy
2. Helps in handling insert, update, and delete anomalies, ensuring a better performance of the database

### Objectives of Normalization

1. To free the collection of tables from undesirable insertion, update and deletion depencdencies
2. To reduce the need for restructuring the collection of relations, as new types of data are introduced, and thus increase the life span of application programs
3. To make the relational model more informative to users
4. To make the collections tables neutral to the query statistics, where these statistics are liable to change as time goes by

### Advantages of a Relational Database

1. Avoids duplication of data
2. Ensures consistency of the data that is stored as records
3. Easier to modify data and data format
4. Easier to insert and delete data
5. Easier to maintain security of data

### Data integrity

1. Is a very essential function of relational databases
2. Ensures that data is accurate
3. Ensures that is data is consistent
4. Is achieved through normalization, defined business rules and validated date

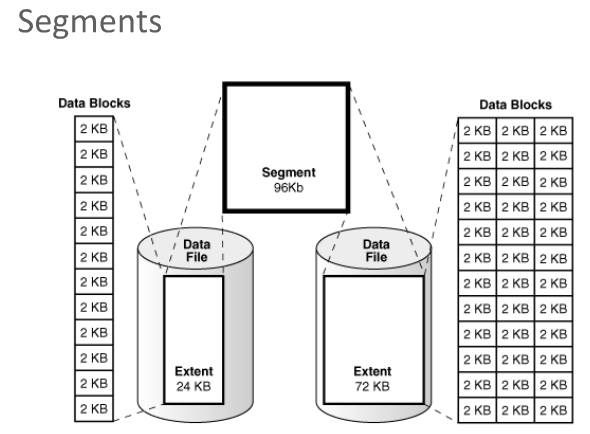
## Hoofdstuk 1.5 Database Storage Structures

### Database Data Storage

1. Is one of the essential task of the database
2. The database has physical structures and logical structures

### Introduction to logical structures

1. Allocates logical space for all data in the database
2. There are four logical units of database spac allocation
   1. Data blocks
      1. Is the smallest logical storage unit of a database
      2. A single data block represents a specific number of bytes on the physical hard disk
      3. The size of a datablock is generally a multiple of the operating system block size
   2. Extents
      1. Is a logical unit of database storage space allocation made up of contiguous data blocks
   3. Segments
      1. Zie foto
   4. Tablespaces
      1. Oracle database stores data logically in tablespaces and physically in data files associated with the corresponding tablespace

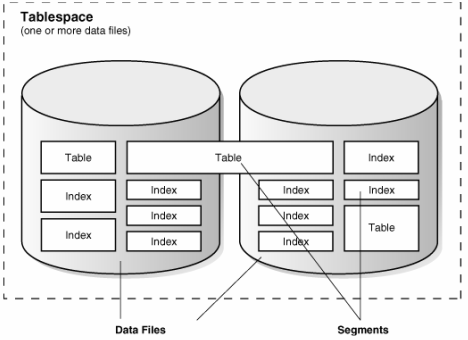


### Introduction to Physical Storage Structures

An oracle database is a set of files that store Oracle data in persistent disk storage

The following files are generated

1. Data files and temp files
   1. Zie foto
2. Control files
   1. The database control file is a small binary file associated with only one database
   2. A control file contains the following types of date
      1. Database name and database unique identifier(DBID)
      2. Time tamp of database creation
      3. Information about data files and online redo log files
      4. Tablespace information
      5. Current log sequence number
      6. Metadata that must be accessible when the database is not open
3. Online redo log files
   1. Every instance of an Oracle database has an associated redo log to protect database in an instance failure
   2. The redo log for each database instance is also referred to as a redo thread



## Hoofdstuk 1.6 Business Requirements

### Why do I need a Database Solution?

1. Mutiple Users
2. Integration of multiple components
3. Muliple data items

### Importance of Business Rules

1. Allow the developer/architect to understand the relationship and constrains of the participating entities
2. Helps you understand the standardization procedure that an organization follows when handling huge date
3. Should be simple and easy to understand
4. Must be kept up-to-date

# Database Hoofdstuk 2 – Databases and Data Modeling

## Hoofdstuk 2.1 What is a Table?

### Single table

1. A flat file database is a type of database that stores data in a single table
2. Flat file databases are generally in plain-text form, where each line holds only one record

### Advantages and disadvantages of a flat file database

#### Advantages

1. Easy to understand
2. Easy to implement
3. Easy to extract information
4. All records stored in one place
5. Simple sorting and filering of reports
6. Less hardware and software requirements

#### Disadvantages

1. Less security
2. Data inconsistency
3. Data rebundancy
4. Cumbersome sharing of information
5. Slow for huge databases

## Hoofdstuk 2.2 Relational Databases

### Relational Databases

1. Presents information in tables with rows and columns
2. Each collums represents a type of information(a field) and each row list one record
3. The tables are related to one another by foreign keys
4. A foreign key is a primary key from a different table

### Advantages

1. Less redundancy
2. Avoidance of inconsistency
3. Efficiency
4. Data integrity
5. Confidentiality

A table is a simple structure where data is organized and stored

### Key terms

1. Column
2. Primary Key
3. Foreign Key
4. Row
5. Field

### Properties of tables

1. Entries in columns are single values
2. Entries in columns are of the same kind
3. Each row is unique
4. Order of columns is insignificant
5. Order of rows is insignificant
6. Each column has a unique name

## Hoofdstuk 2.3 Conceptual Data Modeling

### conceptual modeling

1. Identifies the highest level of relations
2. Contains relationships between entities
3. It does not specify the primary key but it does determine a unique identifier for each entity

### Entities:

Real world object or things that exist indepently examples: person , car, customer

Association between 2 or more entities example: car color 🡪attributes <-- Model

### Creating

1. Identify entities
2. Identify attributes (including optionality)
3. Identify unique identifiers
4. Determine relationships(including optionality and cardinality)

### Entity Relationship model

1. Is a list of all entities and attributes as well as all relatonships
2. Provides background information such as entity descriptions, data types , and constrains
3. It does not require a diagram(it is useful)

### Goals

1. Get all required information
2. Ensure information Is only shown once
3. Model no information that is visible form information already modeled
4. Locate information is a predictable and logical place

### Creating

Create entities and attributes 🡪 choose unique identifiers 🡪 Build relationships 🡪 identify cardinalities 🡪 check the model

## Hoofdstuk 2.4 Data Modeling Terminology

### Table diagram Notations

A table diagram is additional documentation that is often used to further explain keys and collums in the physical database

### Naming conventions for tables

1. The table name is the plural of the entity name examples: STUDENTS becomes STUDENTS
2. Column names are identical to the attribute names, except that special characters and spaces are replaced with underscores

### Naming conventions for Columns

Column names often use more abbreviations that attribute names. Examples:First name becomes first\_name or fname.

### Table Short Names

A unique short name for every table is useful when naming foreign key columns

Create short names based on:

1. Entity names that contain more than one word
2. Entity names that contain one world but more than one syllable
3. Entity names that contain one syllable but more than one character

### Naming Restrictions with Oracle

Table and column names:

1. Must start with a letter
2. Can contain up to 30 alphanumeric characters
3. Cannot contain spaces or special characters but “$,” “#”,and “\_” are permitted
4. Table names must be unique within one user account in the oracle db
5. Column names must be unique within a table

## Hoofdstuk 2.5 Unique identifiers and Primary Keys

### Conceptual Data model UID

1. Is an attribute of an entity that is unique across all instances of the entity
2. Has a non-NULL value for each instance of the entity for the lifetime of the instance
3. The value never changes
4. Is diagrammed with a number sign

### Primary key (PK)

1. Is a set of column or a set of columns that uniquely identifies each row in a table
2. It cannot contain NULL values
3. Is an existing table column or a column that is specifically generated by the database according to a defined sequence
4. Must contain a unique value

### Composite UID and Primary Key

1. Composite UID: a combination of attributes
2. Composite primary key:
3. Is a key that consist of two or more columns
4. Examples: if a product is uniquely identified by its model number and revision code, the combination of the model umber and revision is a compound primary key.

## Hoofdstuk 2.6 Relationships and Foreign Keys

### Relationships

1. Represents an association between two or more entities
2. The lines are either solid or dashed
3. The lines terminate in either a “single toe” or a “crow’s foot”
4. The name from either perspective is printed near the starting point of the relationship line
5. Examples:
   1. EMPLOYEES have JOBS
   2. JOBS are held by EMPLOYEES

### Foreign key (FK)

1. Is a column or a combination of columns in one table that refers to a primary key in the same table or another table
2. Relationships in a conceptual data model are mapped to foreign keys in a database table

# Hoofdstuk 3 – Creating a Physical Data Model

## Hoofdstuk 3.1 Conceptual and Physical Data Models

### What is a Conceptual Model?

1. Captures the functional and informational needs of a business
2. Is based on current needs, but may reflect future needs
3. Addresses the needs of a business (what is conceptually ideal), but does not addres its implementation (what Is physically possible)
4. Identifies important entities and relationships among them
5. Does not specify primary keys

### What is a Logical Model?

1. Includes all entities and relationships among them
2. Is called an entity relationship model (ERM)
3. Is illustrated in a ER
4. Specifies all attributes for each entity
5. Determines attribute optionality
6. Determines relationship optionality and cardinality

### What is a Physical Model?

1. Is an extension to a logical data model
   1. Defines table definitions, data types and precision
   2. Identifies views, indexes, and other database objects
2. Describes how the objects should be implemented in a specific database
3. Shows all table structures , incuding columns, primary keys and foreign keys

### Steps to create a physical data model

Model entities as tables 🡪Model relationships as foreign keys 🡪 model attributes as columns 🡪 modify the physical data model based on physical contrains and requirements

### Conceptual and Physical Models

1. The art of developing, planning and communicating produces a desired outcome
2. Data modeling is the process of capturing the important concepts and rules that shape a business and depicting them visually in a diagram
3. This diagram becomes the blueprint for designing the physical thing
4. The client’s dream (conceptual model) becames a physical reality (physical model)

### Hoofdstuk 3.2 Business Rules

### Business Rules

1. A business rule is a statement that defines or constrains some aspect of the business
2. It is used to define entities, attributes, relationships and constrains
3. It includes rules and policies that define how a business operates and handles its day-to-operations
4. There are two types of business rules
   1. Structural
   2. Procedural
5. Example: All order must have a Contact Number

### Source of Business Rules

1. Top management and managers
2. Written documentation
   1. Procedures
   2. Standards
   3. Operations manuals
3. Direct interviews with end users

### Structural Business Rules

1. Structural business rules indicate the types of information to be stored and how the information elements interrelate
2. These rules can always be diagrammed in the ERD

### Procedural Business Rules

1. Procedural rules deals with the prerequisites, steps, processes, or workflow requirements of a business
2. Many procedural business rules are related to time: Event A must happen before even B.
3. Some procedural business rules cannot be diagrammed, but must still be documented so that they can be programmed later

Additional

### Additional Programming

1. Some rules such as the following, must be implemented by programming them as processes that interact with data:
   1. Any employee whose overtime exceeds 10 hours per week must be paid 1.5 times the hourly rate
   2. Customers whose account balances are 90 days overdue will not be permitted to charge additional orders