CHAPTER 1

Midterm Review

End users → sophisticated/casual users:
- * query language
- * enter query
- * authorized by DBA To view part of the data (and manipulate it)
  - Retrieve
  - Insert
  - Delete
  - Update

QUERIES

DevOps

USE DATA TO ACCOMPLISH THEIR TASKS

→ Data scientist → STATS & QUERY
- * analyze large volumes of data
- * query language
- * analytic and reporting tools
  - find patterns
  - find relationships
  - find trends

→ Secondary users

use information without interacting directly with it

→ Naive users → APP
- * DO NOT USE INTERACTIVE QUERY LANGUAGE
- * invoke programs by entering simple commands or choosing options from a menu

Registrar clerks

The Princeton Review

Application programmers

Languages = Java, C, C++, C#, R, Visual Basic, or PHP

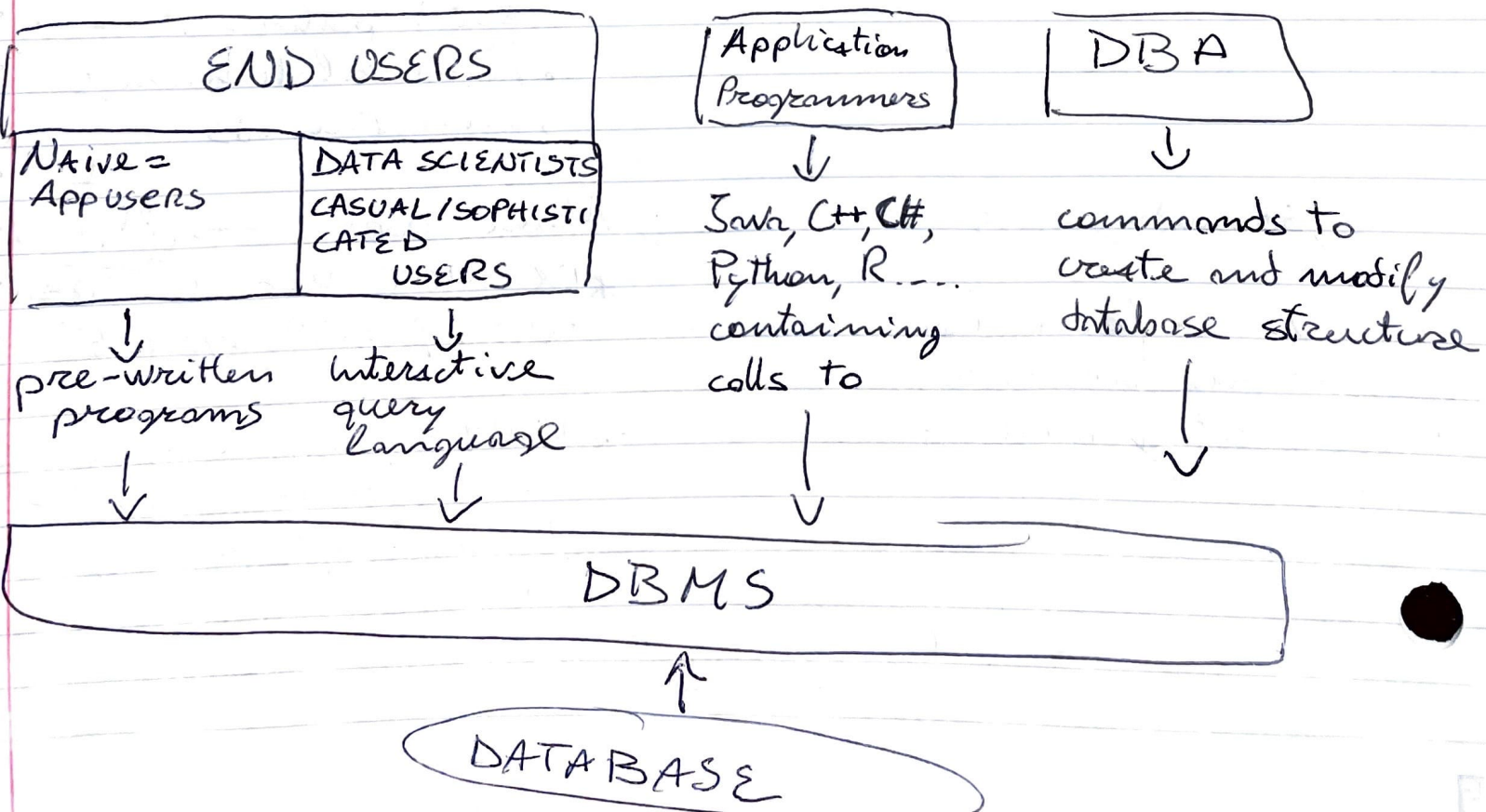
(sophisticated users) calls to DBMS

DBA = Responsible for designing, ~~creating~~ structuring, securing, and maintaining databases

Specialist → designs ~~creates~~ → DB

DBA → implements ~~MANAGES~~ → DB

* Data engineers work closely with data professionals who work closely with administrators.



Integrated Database Approach Benefits

Unlike partial information typical of file processing or small databases, integrated database solutions allow for:

* Sharing data → Enterprise authorizes who may view information
* redundancy control → fewer copies of the same data
* data consistency → updates to data are consistent across the database
* improved data standards: → rules to enforce formats, conventions, documentation....
* (Result) Better data security: → Protection from unauthorized access to data: MASKING v ENCRYPTION
* Improved data integrity: not permitting record insertion, deletion, or update that conflicts with integrity constraints
* Balancing of conflicting requirements: DBA needs to find a solution that best suits everyone's needs.
* Faster App Development: Data inside DBMS is already available in the desired format for App Development
* Ease of access to data: use of query languages to access data
* Scalability: replicating one successful database model with pooled resources proves cost-effective in LR.
* Control over concurrency = one user viewing data and another one modifying data should not interfere with one another's operations.

* More reliable backup and recovery procedures =

records are normally backed up (copied and saved) on a regular basis.

Updates are stored in a log of changes called recovery log.

If the system goes down, an organization may retrieve their DB and data from backups and recovery log.

Historical breakThroughs in Information Systems

DBMS addresses the need to preserve data

1890 → Punched Cards → census ~~record~~ records
(Herman Hollerith)

1940s → ~~paper~~ punched paper tape

1950 → magnetic tape → INPUT FOR EARLY COMPUTERS
↓
UNIVAC I

paper tape, punched cards, and magnetic tape could only be read in the order it was stored. (sequential file sharing)

→ batch processing efficient ↙ ↘ NOT FLEXIBLE

Payroll example:

*sequential

NEEDS TO BE IN THE SAME ORDER

Payroll master file: #employeeID1, #employeeID2, #employeeID3, ---

Transaction file: #hrswID1, #hrswID2, #hrswID3, ------

||

Payroll program

↙ Paychecks and stubs

↓ Payroll reports

↘ New payroll master file

old master/new master or parent-child system

Late 1950s → Magnetic Disk → made direct access (nonsequential access) of records possible.

Pain addressed:
programs no longer required that the order of access match the physical order of the records

Benefit(s): updates saved to disk instead of rewriting an entire file

COBOL
PL1 → commercial data processing

Drawbacks: moving parts / slow speed $

1970s → Solid State Drives → electronic storage devices

↓ were not available in large scale until the 2010s.

↓ NO MOVING PARTS

↓ FASTER ACCESS TO DATA

DATABASE MODELS

1960s → hierarchical model (IBM's Info Management Sys 1968)

tree-like structure: it contains nodes that are connected by branches, and the primary node is called the root node. Each node has exactly one parent, but one parent can have many children.

Early 1960s → NETWORK MODEL: Bachman's Integrated Data Store (IDS) — CODASYL model

↓

data is presented using directed graphs. Thus, data nodes can be interrelated, and relationships can vary from 1:1, to 1:MANY, MANY:MANY.

COBOL language

The hierarchical and network models were powerful and efficient, but they were complex and required users to know data structures and access paths to data.

↙ ideal for programs

↘ NOT MEANT FOR INTERACTION W/ USERS

The industry decided to adopt a user-friendlier solution

↓

Relational data model

1970 ——> Codd proposes The relational model

* based on abstract concepts of math
* development of a new language, SQL (Structured Query Language)

Larry Ellison and Meyer and Oates founded Oracle in 1979.

↓

Oracle uses The results of The System R project

↙

first Relational Data Base Management System

SQL based Relational DBMS's ~~language~~ used nowadays:

Oracle, DB2, SQL Server, Access, PostgreSQL

↑ ↑ ↑ ↑ ↑

Oracle IBM Microsoft Microsoft open source

Relational Database uses simple tables to organize data but offers little flexibility in expressing important decisions.

1976 —> Peter Chen —> Entity - Relationship (ER) model

Benefits: ~~easily~~ captures The semantics of data
easily store and manipulate data
friendly to Object-Oriented Programming

(unlike in RDBMS) ↗

Data warehouses are a method of capturing data consolidated from many databases. A data warehouse usually stores ~~historically~~ historical data about an organization to perform analytics.

Data Lakes are repositories that store data in their natural format. The raw data is transformed when needed for use in data analytics.

Internet —> uses —> semistructured data model —> for differently - structured data or relatively unstructured data stores.

CHAPTER 2:

STAGES in DATABASE DESIGN

Determine current user environment (inputs/outputs) (how they use system → consider all needs

Using STEPS SKETCH CONCEPTUAL MODEL ← IDENTIFY ENTITIES, ATTRIBUTES, AND RELATIONSHIPS
ABSTRACTION → FIND COMMON PROPERTIES → TO CATEGORIZE DATA

Choose the DBMS that best satisfies the target organization

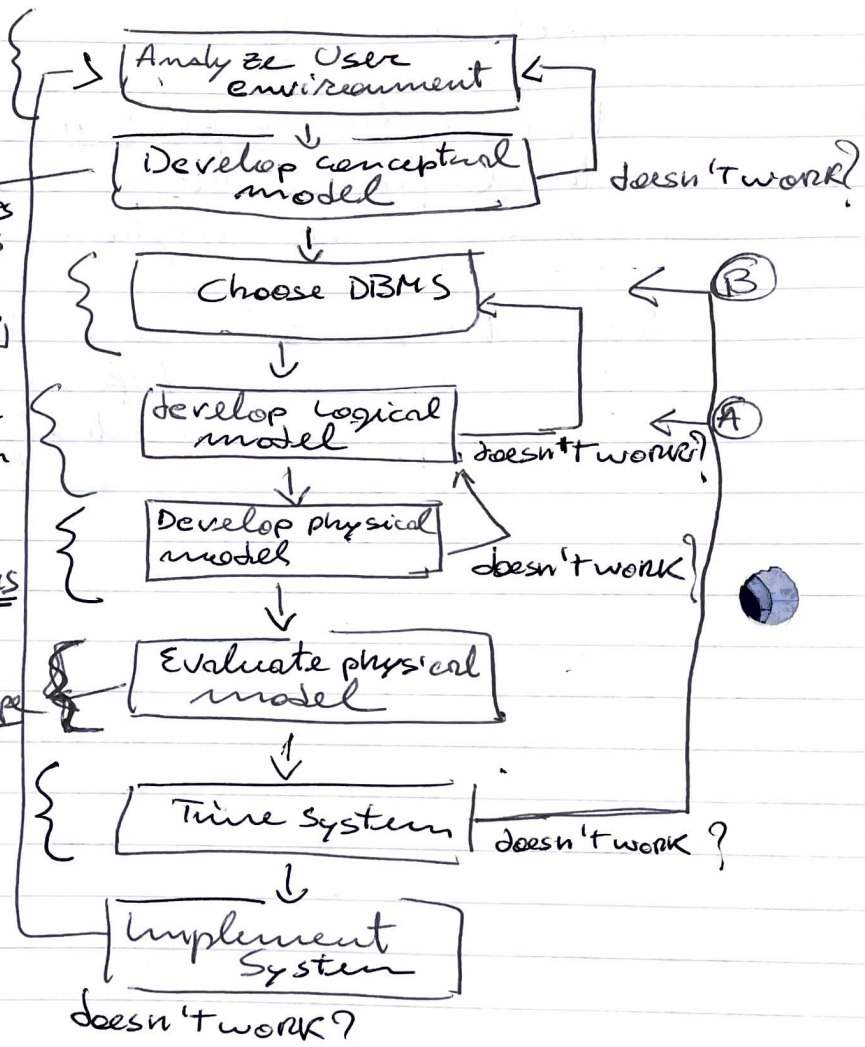
mapping the conceptual model to the database model used by the chosen DBMS

laying out the data considering the structures supported by the DBMS

Testing the performance of all applications, transactions, and quantitive data through a prototype

modify physical structures or optimize software

if the model passes the evaluation, the designer may implement it adding the security subsystem



ALL CATS DOGS LIZARDS PAINT TRAINS

INDIGO

DESIGN TOOLS

* DATA DICTIONARY ≐ Active Data Dictionary ≐ Integrated Data Dictionary.

DATA DICTIONARY is a repository of information about the logical structure of the database.

It contains data about the data in the database, also known as metadata.

There are also data dictionary tools without a particular DBMS called freestanding data dictionaries.
Ex: commercial product, txt file, or spreadsheet.

Many prefer freestanding dictionaries early in the design stage before choosing a DBMS solution. (flexibility)

Once the database is created, it is hard to make adjustments to the ~~database~~ data dictionary, and over time, the dictionary will not be an accurate reflection of the database structure

freestanding dictionary's benefits:

* collecting and storing information in a centralized way
* securing agreement from users and designers about the meanings of data items (~~[illegible]~~ agreeing on definitions)
* communicating with users (identifies user per item of interest)
* addressing redundancy and inconsistency
(synonyms and homonyms)
↓ different names for same item
↓ same name for multiple items

* Keeping track of changes to the database structure
* Determining the impact of changes to the DB structure
* Find sources of and responsibilities for the correctness of each item

capture data near to its source.

DBMS system catalog → DATA DICTIONARY + DBMS performs the above operations and more:

* records external, logical, and internal schemas and the mappings between them.

* records info about users authorized to access certain items
  ↳ enables retrieval, insertion, update, or deletion

* provides audit information → spot security violations

Diagramming Tools: Visio - - -

Computer-aided Software Engineering (CASE) Packages

analysis ↙ project management ↓ design ↓ reports ↓

Database Administration

design ↙ ↓ operation ↘ management.

implements design, develops the system, and manages it

1) Planning and Design

* Preliminary planning (investigation and feasibility study)
* Identify user requirements (information needs)
* developing and maintaining DATA DICTIONARY
* SKETCH/DESIGN CONCEPTUAL MODEL
* CHOOSING DBMS
* Develop logical model
* Develop physical model
* Develop security plan

2) Developing the DB

* Installing the DBMS (Oracle)
* Creating and loading the DB
* Securing the DB (authorized access)
* Developing user views (~~DB for each access~~ SATISFY USERS)
* Writing and maintaining document.
* Developing and enforcing data standards
  (format, acceptable ranges of values, uniqueness)

* Developing and enforcing application development standards: (audit)
* Developing operating procedures (periodic checkups)
* Train users (teach to use it effectively)

- MANAGEMENT ~~DATABASE MANAGEMENT~~

* Ensuring DB SECURITY (identify vulnerabilities)
  MASKING vs ENCRYPTION

* MONITORING PERFORMANCE
  statistical analysis on performance of DBS
  (running time, response time)

* Tuning and Reorganizing (changes to store data)

* Keeping current on database improvements
  (is it worth it upgrading to a new system?)

DATA MODELS

Data model = description of the database structure

↓

includes → data
→ relationships
→ constraints

ER-model = semantic model type → conceptual level of data

* identify entities, attributes, entity set

↓

collection of entities of the same type

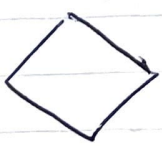
students → student

Relationship set → set of relationships of the same type

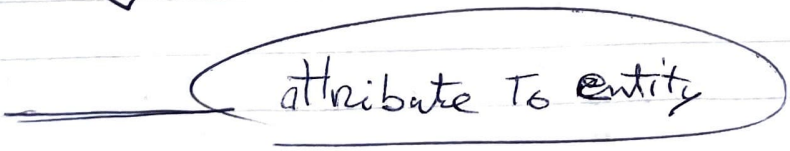
descriptive attributes
* joined_date
* grades

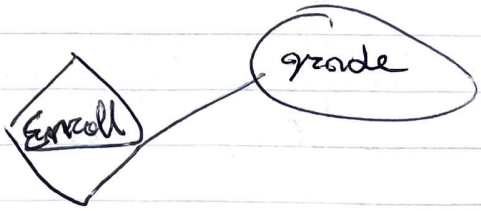
entity set

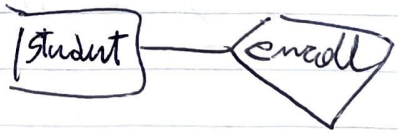
attribute

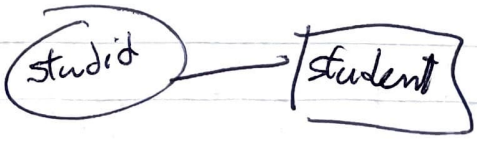


relationship

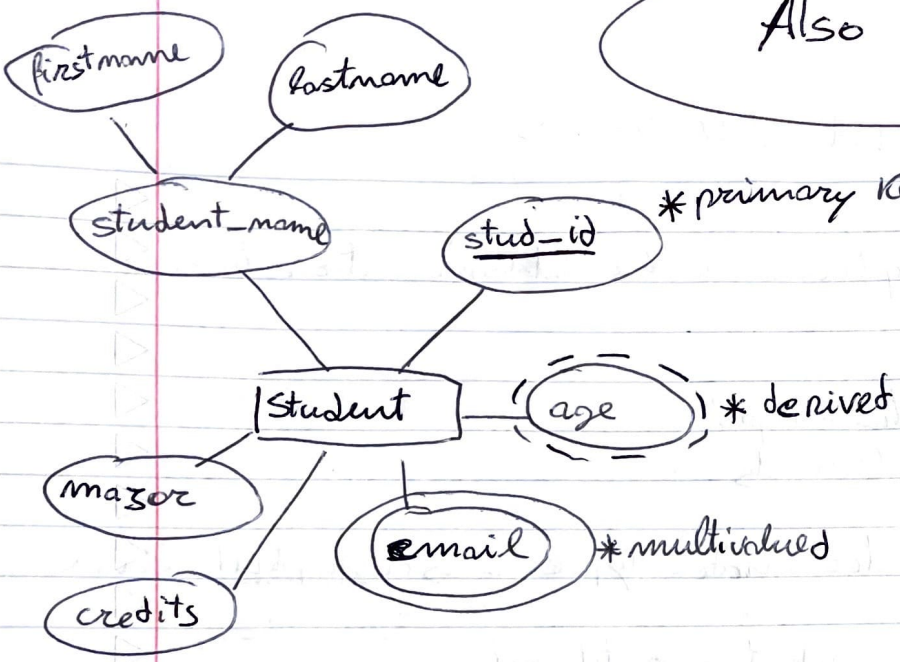








Also point 3.4



Domain of the attribute = set of allowable values for attribute for instance 0 to 150.
Attributes may have null values

Multivalued att = some attributes may have multiple values (email)
Composite att = decomposing complex attributes into smaller attributes
SuperKey = attribute/set of attribute that uniquely identifies a row ~~for~~ a table

Candidate Key = is a superKey w/ no redundant data

Composite Key = a key with two or more attributes that uniquely identify a row in a table

Alternate Key = candidate key that is not chosen as primary key

Secondary Key = attribute/set of attributes to access records

Foreign Key = a key that references the primary key of another table

Relational model = record-based model
↳ picturing what the logical records look like.
* allows designers to develop and specify the logical structure

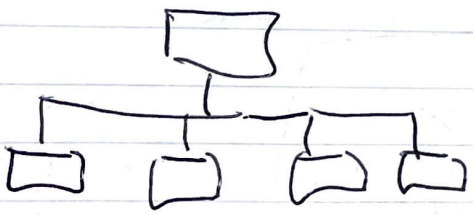
DO NOT PROVIDE SEMANTIC INFORMATION SUCH AS OBJECTS, RELATIONSHIPS, ABSTRACTIONS, OR DATA CONSTRAINTS

* USE OF MATHEMATICS.

* ENTITIES and RELATIONS AS TABLES AND ATTRIBUTES AS COLUMNS OF THOSE TABLES

* hierarchical and network models are also record-based models

hierarchical



network

Object-oriented model = primarily semantic model

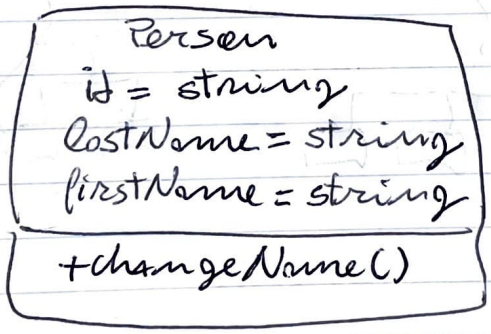
objects, like entities, represent people, instances, anything that can be cataloged.

object has a state and a behavior
↓ ↓
determined by the values of the attributes | set of operations called methods

A class = entity set and consists of the set of objects having the same structure and behavior.

* Encapsulation = data + function in a unit where they are protected from outside changes

Ex:



Data Warehouse Models ——> support analytical queries (Analytics)

* star schema is the data model for large, static data in a d. warehouse

- 1 single central Table called fact Table
  each attribute has its own table, or a dimension table

- store data using multi-dimensional arrays called data cubes or hypercubes

  * pivot function
  * rollup to aggregate and combine data
  * drill-down to obtain more detailed information

Semistructured data models: existing databases having different schemas

* documents (XML/HTML)
* JavaScript Object Notation (JSON)
  ↓
  human-readable represent data as attribute-value pairs
  Values may contain objects and arrays