Lect 1
INDADD

Introduction to Database Design and Development

Get early feedback on coursework.

- Relational Database Systems

Outcomes

- Demo structured analysis techniques to elicit requirements for Business IS
- Identify design methods Bus. IS
- Utilising SQL
- ID legal, ethical, professional issues regarding Bus IS

Sally Bus. (syllabus)

- IT/IS impact on Business
- Bus IS dev. and project lifecycle.
- Sware dev. cycles.
- Sys Analysis Tools + techniques
- Database design, ERDs
- Database environment
- db dev. and Implementation.
- SQL
- Sys testing Methods + Sec.
- Delivery + Installation.

Software :- Nearpod (lecture interactivity)
- Virtual machine.
- MS Visio.
- My SQL
- My SQL workbench.

Scanned by CamScanner

Assessment

80% - exam at end of year

70% - Coursework Portfolio - Start early.

- 3 Submissions over year.
- Draft review
- Marks within 20

What is a database:

Single repository of data that can be used simultaneously by many departments and users.

Relational database:

- Data represented as a table.
- Each row of a table represent one record.

Database Management System

- Defines and regulates the collection, storage, management and use of data within a db environment.

- Oracle db and MySQL ↑ transferrable.
- Redundancy.
- Centralization.

Big Data: Volume, Velocity, Variety, Veracity, Value.

Technologies.: Hadoop, NoSQL, NewSQL

Scanned by CamScanner

LECT 2

INDADD - DB Environment Introduction

Relational databases w.th SQL

↳ Implementation of databases.

Data : Raw eg 'Red', 192..., v2-0

Information : Meaning/~~context~~ : Traffic light turns red

Knowledge : Context : I am driving towards a red traffic light

Wisdom : Application : I should stop the car.

DB use eg : Satnavs, libraries, supermarkets, online order fulfilments, travel agents, credit card management.

Management IS

Processes Data into High Quality Information.

~~Process, moves, activities~~

- Data Collection / storage
- Transformation (queries)
- Dissemination (Network/UI)

Business Processes

- Supply chain management
- Enterprise, customer, knowledge management

↕

Management Activities

- planning
- coordinating
- Modeling

"Group of interconnected subsystems working together to achieve one goal."

Scanned by CamScanner

Management Information - Information provided to management members to make decisions.

Management Information System - means by which the right piece of information is delivered to a decision maker at the right time and in the right format for effective decision making to take place.

Placing data in a meaningful context helps reduce uncertainty.

- Create files to contain data
- File data stored in records (rows)
- Structured with fields.

Issues with file system
- Changing file format requires each program to be changed
- Conversion and compatibility issues.

Data base solves this
Let programs access data through a DBMS.

Schema, Maintainance and Utility Programs.

Scanned by CamScanner

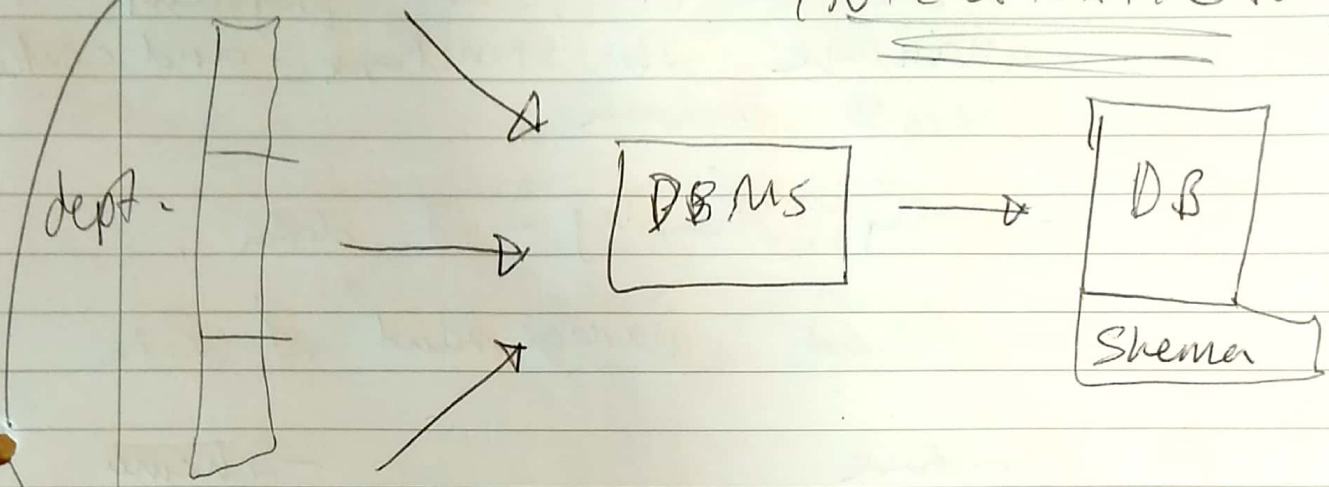
WEEK 3

INDADD: Database Concepts

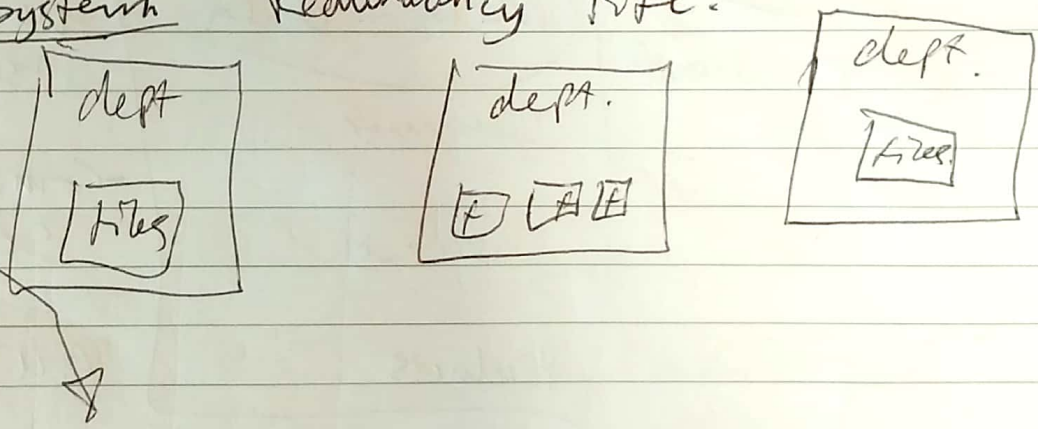
→ relational.

db system

INTEGRATION!



File System Redundancy Info.



Adv – db centre of interest
– data maintained independantly.
– general maintenance and utility programs may be deployed.
– DATA SEPERATED FROM APPLICATIONS

– Data must contain descriptions of data
↳ Metadata / schema, information about data ↳ relation between dbs

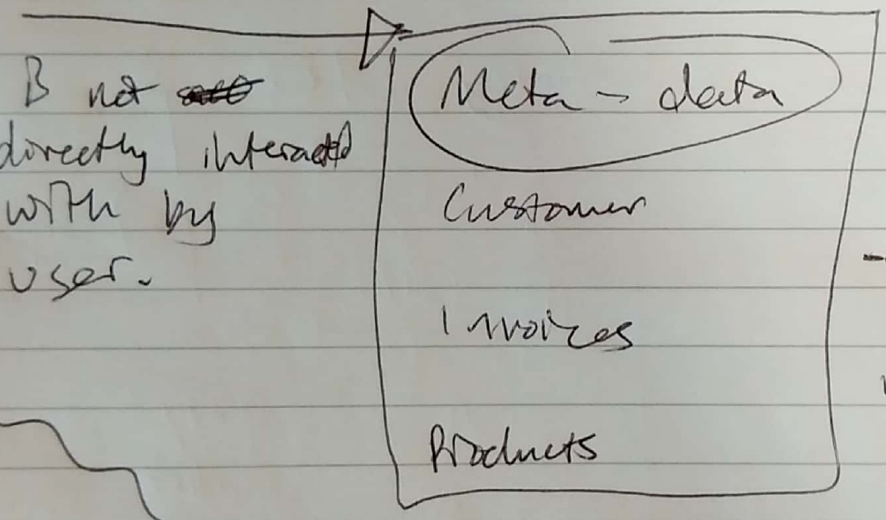
– Need a way to logically group objects, tables, views, stored procedures.

Scanned by CamScanner

Database MS (eg. My SQL)

- collection of / one program to manage db structure and control access to data.
- easy sharing of data.
- efficient management of data

db structure



- Schemas stored with the database itself.

- Communication via SQL or via GUI

Adv: - End users have better data access.
- Integrated view of operations
- Data consistency for higher
- easier to run network queries.

db types:
- Single user -
- desktop - Single user on PC
- Multi user -
- work group - Multi user, single dept
- Enterprise - Multi user, whole organisation.

Scanned by CamScanner

db Classification :

Location : — Centralized
— distributed

Use : - transactional, daily operations. ★ Fast.
- data warehouses - Stores data.
↳ query and analysis ★ Slow
↳ store historical data.
↳ different structure.

dbMS dbs:

- Users should be able to access same data.

- User view immune from changes in other users view

- physical db storage details not need to be known

- DBA

- data independence. changes in storage structure and data access technique doesn't affect application program

Three - Level Architecture.

- Specific fields — External level : User View
- all fields — Conceptual level : Conceptual Schema
- SQL — Internal level. : Internal Schema.
★ physical data organisation.

Scanned by CamScanner

- data Independence: Physical / Logical.

db Languages:

Data definition Language DDL:
ALTER - creation of objects (tables, views)
CREATE - dealing with objects.
DROP - creating entities and their relations.
RENAME

Data manipulation language DML:
- creating new records.
- dealing with the data inside objects.

Creating a table

```
create table <table Name> (
    -> fields data type NULL/Not Null,
    ->  "      "     "    "
    ->  "      "     "    "
    -> define primary key );
```

cannot Select without From

db development life cycle.

- Planning
- Collect requirements
- Analysis
- design
- prototyping
- Implementation
- ... Maintenance (cover in more detail later)

Scanned by CamScanner

Wk 4

LNDADD: The Front End

~~Used~~ used to interact w/ a db

systems - people are a part of the consideration.

Problem Definition

- Idea and Objective.
- Size and boundries.
...

Feasibility Study

- Economic feasibility — costs — comparison.
  - benefits.
    - more than financial.
- Technical "
  - Available resources.
- Operational "
  - organisation?
    - will it work within?
  - people reaction?
  - business opportunities
- Schedule "
  - how reasonable project timetable is.
    - take too long?
  - Organisational deadlines.
- Legal - contractual "
  - project legally sound?
  - Data about people and money?
  - Govt regulations?
  - Software ownership?
  - security?

p.7:
- possible
- practical
- desirable

Scanned by CamScanner

- Political " → System supported by Management?
  ↳ System supported by Staff?

⇒ Feasibility Study summarised in a report
  ↳ Justifies whether or not the project should go ahead and when.

- What do you want to know before spending large sums of money on the development of a new system?

- Answer every question answerable on a project.
  ↳ requires project Management.
    ↳ ~~Ends~~ goes until the delivery of a project.

Project Failure: - Financial cost
  - cost to Morale
  ↳ Often due to poor project development and Planning

Project Needs - Specificity
  - distinct instructions and finish.

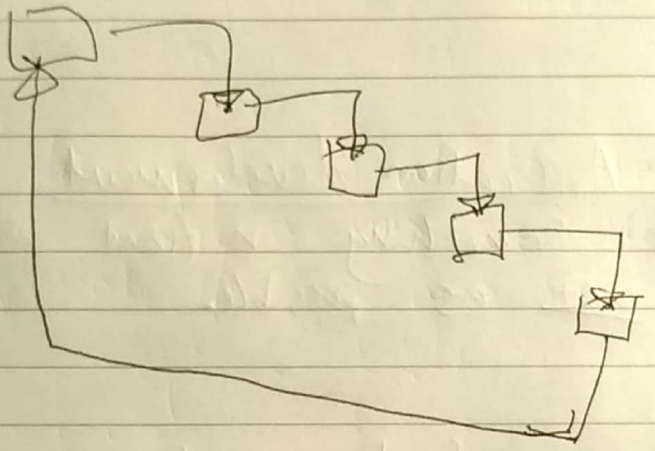
SDLC - Systems development Life Cycle: a framework for the process of System development
(What to do, and when.)

Scanned by CamScanner

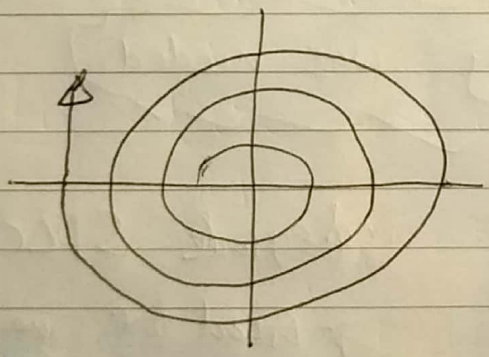
System development processes.

Implementation – coding and creating artifacts
↳ of design.

- 1970 – Royce – Waterfall model.



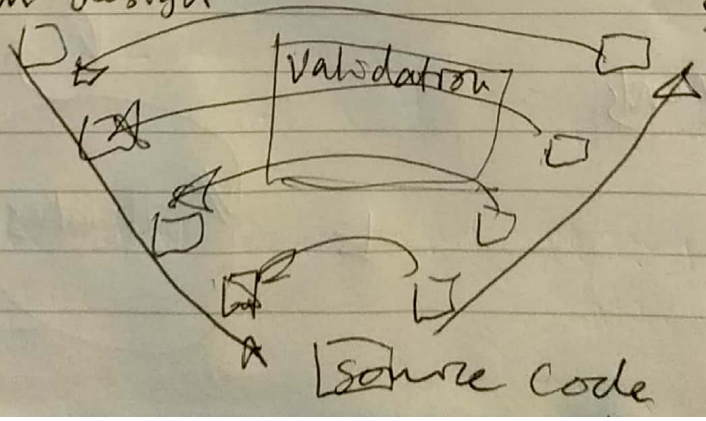
- 1986 – Boehm – Spiral Model.



Risk analysis each iteration.

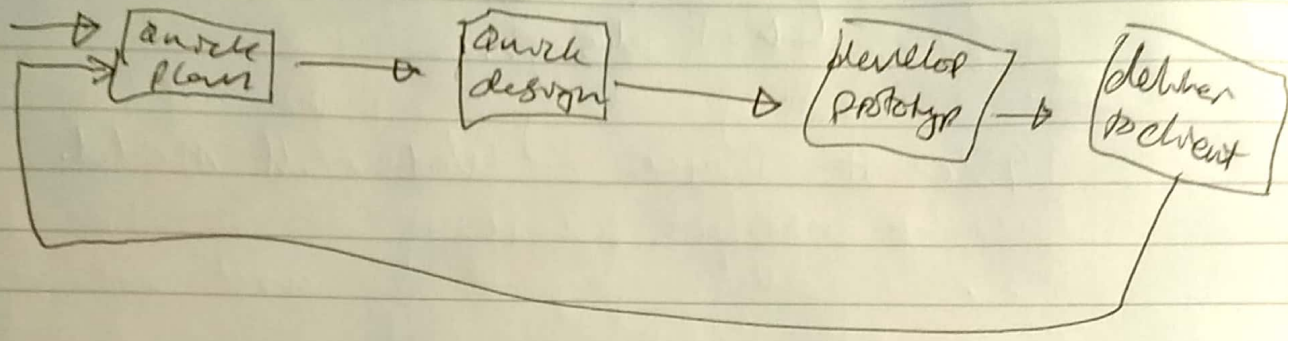
- The V model. – Testing as a key component.

System design



Scanned by CamScanner

Prototyping Model - direct relation and comm w/ client during development



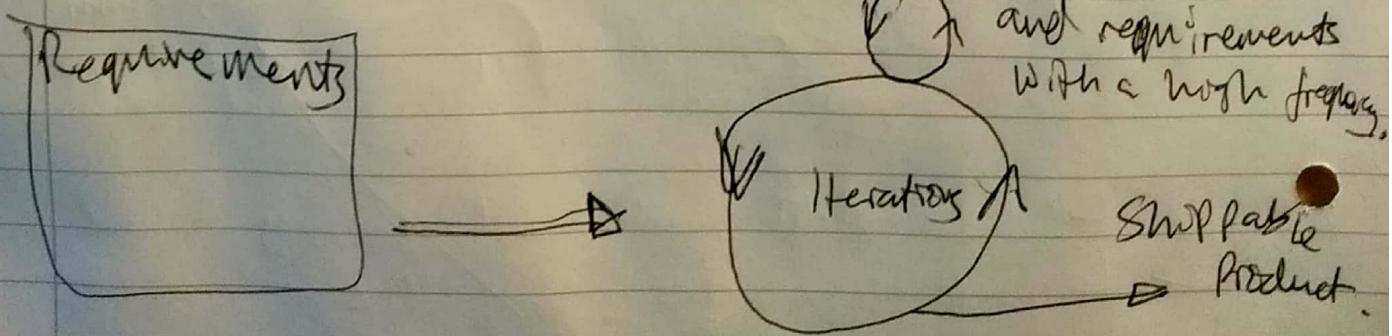
- Rapid Application development
  - ↳ get something in front of the client as fast as possible.

Key obj.
- ↳ high speed, quality
- ↳ low cost

but flexibility was needed

* Agile Methods integrate SDLCs and PM systems *
  - Good for time sensitive projects.
  - fast

Agile Manifesto:
The Agile Approach



Scanned by CamScanner

INDADD : Systems Analysis Techniques.

CW submissions

1 - DFD - 15 Dec - 10%
2 - ERD - 2 Feb - 12%
3 - SQL - 29 Mar - 38%
start early. 60?

Feeds into a data flow diagram

Analysis

Fact finding techniques - how do you gain information.

Who is involved

Internal / External / Operational / Executive Stakeholders.

Analysis → Internal: Staff → Broken down by functions.
Stakeholders → External: suppliers, customers.
→ Operational: DBA
- Executive: Senior Management.

Examining Documents: What might be reviewed?
- reports, memos
- email
- invoices
- order forms
- customer / supplier files
- price lists
- procedure manuals - flowcharts
- data dictionary.
- program documentation
- Publicity materials
- Information booklets.

Helps understand data and data structures and understanding data procedure.

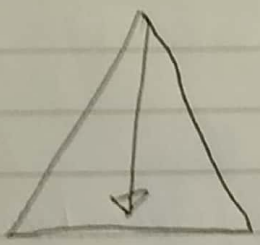
Scanned by CamScanner

Interviewing

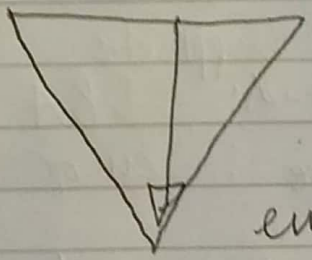
- Objective / purpose
  - to know the type of data you need.
- W? W? W?
  - place, time.
- Plan structure
- Plan questions.

Interview structures

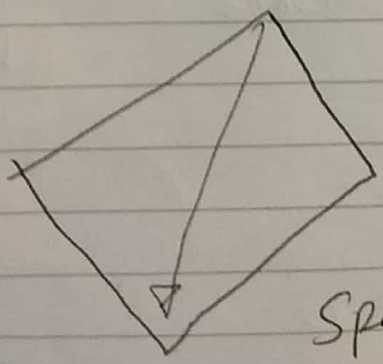
Pyramid



Funnel



Diamond.



Scanned by CamScanner

Asking Questions

- Functional Qs
- Operational Qs
- Business Qs
- Implementation Qs

Interview Procedure

- Introduction
- Consent for recording.

- Vary question type to gain different types of information
- Clarify and understanding
- collect sample documents
- Maintain relevancy.

- Consider further appointments.
- write up notes.
- Allow interviewee access to notes and recording → DPA.

Observation and Shadowing

- Direct
- Shows what people are doing
- View activities.
- Reveals informal / undocumented flows
  - Hawthorne effect.

Scanned by CamScanner

/ fast

Questionnaires

- Low response rate
- Simple → provides little information
- Complex → leads to misunderstanding.
- Useful when
  - When people are geographically scattered
  - Large numbers.
  - When an overall opinion is needed.
- A good one would
  - be simple
  - Clearly set out
  - Language and understanding
  - benefit respondant.

Administering Questionnaire?

- Hand out forms → get someone else to do it.
- Online - email.

Selecting who should answer the Questionnaire?

- Representation of the whole.
- Determine population
- Less sampling required online.

Scanned by CamScanner

Wk 6

INDADD - Data Flow Diagrams.

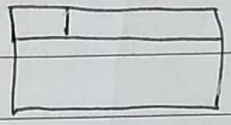
↳ Find out how data flows through a system? -> origin -> capture -> process -> storage -> distribution.

↳ produced during analysis stage.

↳ refined during design stage.

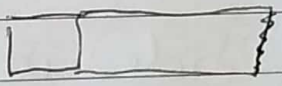
→ comm tool.

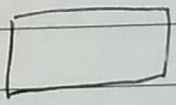
4 symbols

Process 

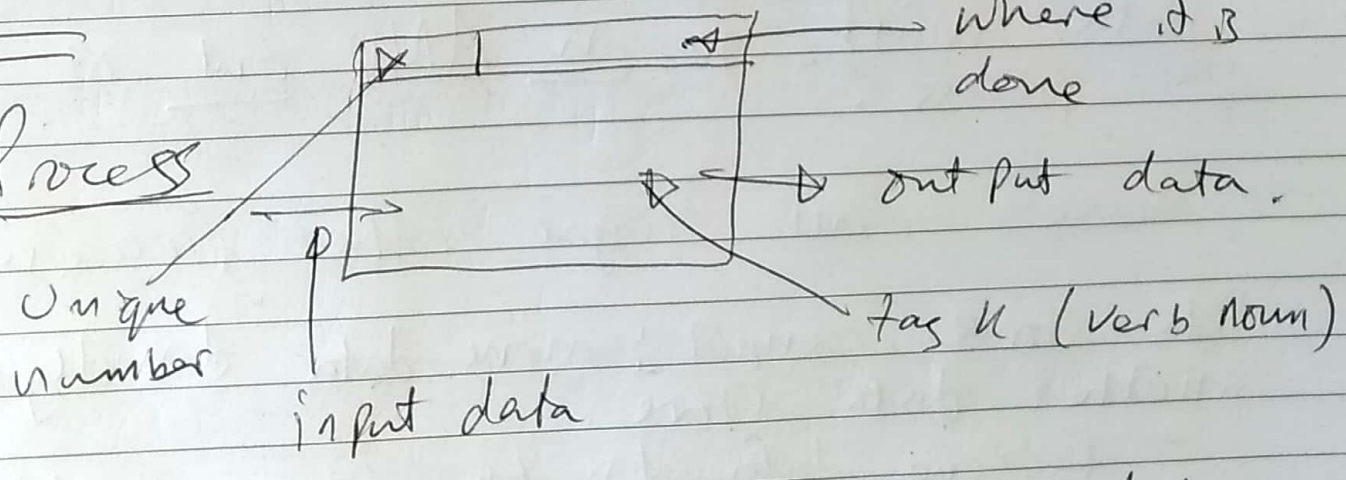
Gane and Sarson

Data flow →

Data Store 

Sources and Sinks 

Process

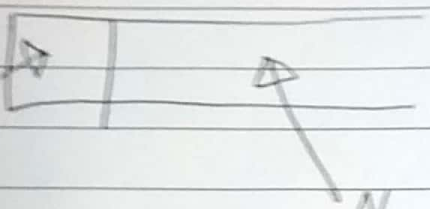


Dataflow - Arrow w/ text describing data

data. (noun) →

Scanned by CamScanner

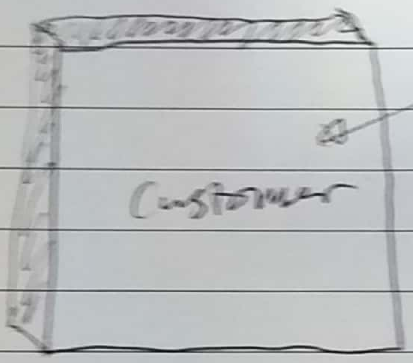
Data Store



Unique ID

Name describing the data.

External Entity



Name describing the source/sink.

"People out side of the System"

Rules

- Data must go in AND out of a process.
- Process can trigger other processes.
- Data stores cannot: move data directly to another data store.
  - Receive data directly from sink/source
  - give " " to " "

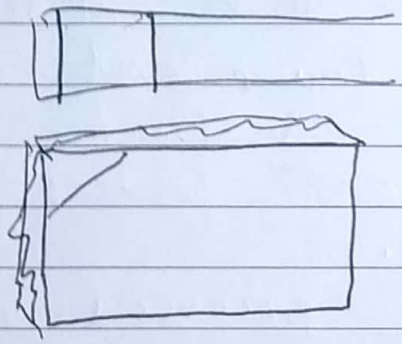
External entities cannot directly interact.

- Data only flows along one needed arrow

⇒ Data cannot feed into the same process

Scanned by CamScanner

- Like to denote a duplicate.



How Much detail?.

- ↳ Impossible for complete representation.

- Start with an overview of the business and continues by analysing each functional area.

Levels - Context diagram - Level 0

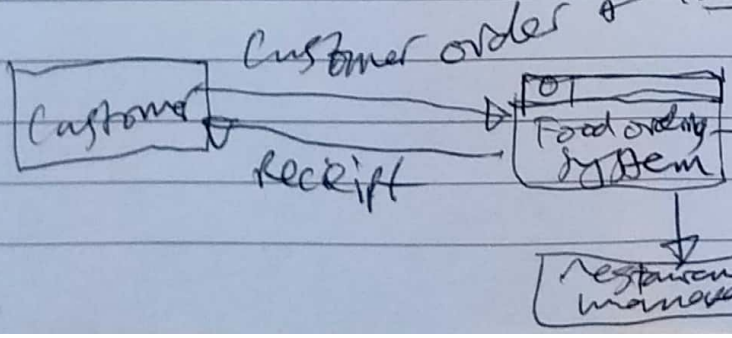
- Level one - 1, 2, 3
- Level two - 1.1, 1.2, 2.1, 3.1

breaks down processes until system decomposition.

Lvl 0

- Describe overall process
- Identify external entities.
- link with data flow

beware of too much detail.



Scanned by CamScanner

Lvl 1 - Diagrams

- Describe main functional areas.
- Avoid lower level processes.
- Start construction with level 0
- Lay out sources and sinks and data flows.
- ID the processes that make up the context of processes.
- Draw in any data stores used.
- Link new processes.
- System boundary, what's in and what's out.

Scanned by CamScanner

external / internal

The Tutor Marked Assignment System (eTMA)

The eTMA system allows students to submit their answers to tutor-marked assignments (TMAs) as computer files to the University via a website. Whenever a TMA file is submitted, it is stored in a central database and a 'receipt' is sent to the student to acknowledge that the TMA has been received. Tutors are informed, by email, that a TMA is waiting for them to be marked.

The system enables tutors to download their students' submissions, mark and comment on the assignments 'on-screen' and submit the marked TMAs back to the University. A marked TMA is stored in a database and the student is informed, by email, that their TMA has been marked and is available to be retrieved electronically.

When the tutor downloads an unmarked TMA, he/she also receives an electronic version of the PT3 form on which the marks awarded for each question and the overall comments on the TMA must be entered. The completed PT3 form accompanies the marked TMA when it is sent to the University's database, and eventually both of them are sent electronically to the student.

LvL 1 starts with the external entities.
LvL 0/1/2 has the same I/O for external entities.

Scanned by CamScanner

Wk 3

INDADD Further Requirements Documentation

Data dictionary - a db that contains all the data and system definitions for one or more databases.

↳ Describes structure and attributes of data items.
↳ ensures conformity
↳ in hands of DBA

Data Dictionary: ↳ Pareto documentation
→ eliminates redundancy
→ Validates data flow diagram

element = field, smallest piece of data that has meaning.

record. - meaningful combination of related data elements included in a data flow, retained in a data store,

↳ requires: - Name and label
a few different things: - Alias
- Type and length
. . .

Data dictionary is an absolute Must!
↳ lots of benefits.

Scanned by CamScanner

Decision Tables: - Models Logic
- Made up from: conditions, actions and rules.

conditions and actions	Rules	
	1	2
< £200	Y	N
> £200	N	Y
Put through Sale	X	
Request check.		X

Process for developing decision tables:

1 - Determine number of decisions

2- Determine possible number of actions

3 - Determine the number of condition alternatives.

4- Insert X

Scanned by CamScanner

Withdrawing money.

Cond	Rules							
	1	2	3	4	5	6	7	8
Requested amount < amount in account	Y	N	Y	N	Y	N	Y	N
PIN Entered correctly	Y	N	N	Y	Y	N	N	Y
~~output~~ overdraft available	Y	N	Y	N	N	Y	N	Y
output cash	X	–	–	–	X	–	–	X
refuse	–	X	X	X	–	X	X	–

☆ Not needed – is a separate process with a separate decision table.

Also there are trees;

- If all the documentation was given to a developer, could they create the system?

WK 2

INDADD - ERDs

OBJ - Link Analysis with design
- Int DD
- Int: - Business rules
- Entities + relationships
- Business rule notation.

DB dev.
↳ 1 - Extract Business Rules
2 - Design
3 - Implement.

Business rules: look at the Mandatory, optional, User, Data requirements.
↳ behind rules exist the constraints on how these tasks are adhered.
↳ Underpins business's structure and how it operates.

ERDs - Entities: Nouns
- Relationships: Verbs.

Business rule Notation
Many - M, *
one - 1
Zero - 0

ERDs can be built up from a list of relationships. what to them apply not for logic

A table in a relational db is called a relation.

Scanned by CamScanner

Properties of relations
- 2D structure.
- row represents a single occurrence.
- Unique field names. — Throughout the db.
- Cell contains one specific value.
- Values in a field must be in the same format.
- Combination of attributes will identify a row.

Relation - Schema. - Looks at the attributes (correct table?, correct order?)

Primary key - Unique value that identifies a tuple.

Several types of keys in use
- P. keys from another table are foreign keys.

=

Candidate key - Columns that CAN uniquely ID a tuple
Primary key - Uniquely IDs a tuple.

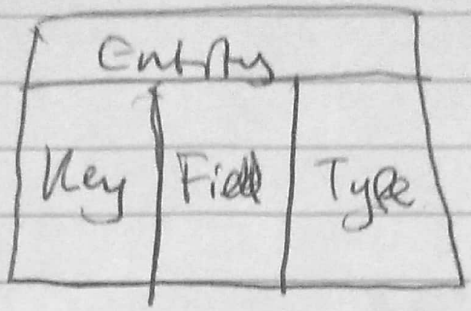
Composite key - Unique identification via multiple fields.
Foreign key - primary key from another table.

1..M - ideal relationship, used often
1..1 - Used rarely
M..M - ~~Used~~ not used in a relational db.

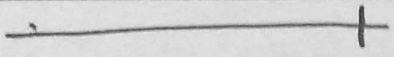
Scanned by CamScanner

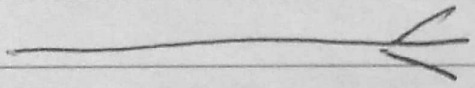
ERD

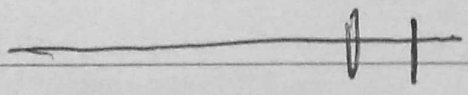
Entities

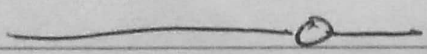


Relationships


one


many


one and only one


zero.

Scanned by CamScanner

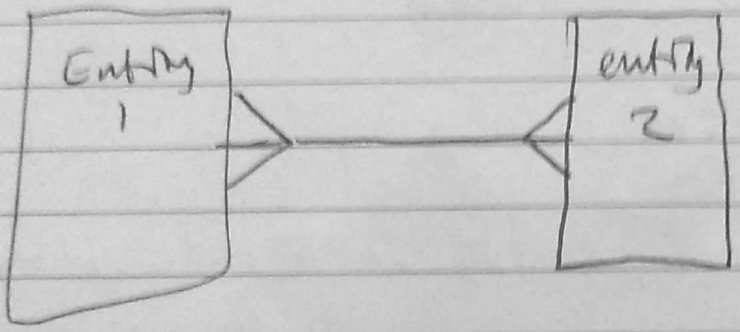
Wk 10

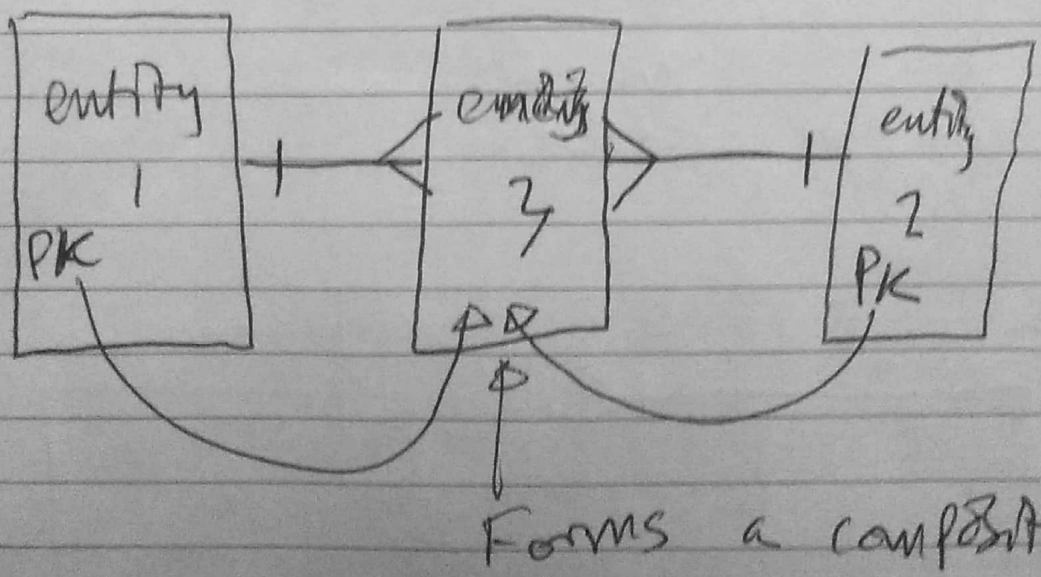
INDADD - More ERDs.

building up an ERD.
- Nouns - entities
- Verbs - relationships.

normalising a database - make sure there is only one ~~entry~~ record

- Many to Many relationships not allowed in relational dbs
  - relationships must be "resolved out"





Forms a composite PK.

relation = table

Scanned by CamScanner

Degree of a relationship
↳ number of entities involved in the relationship - usually 2

ERD Process - Create list of tables and attributes
- Create an ERD to show how you would track this information

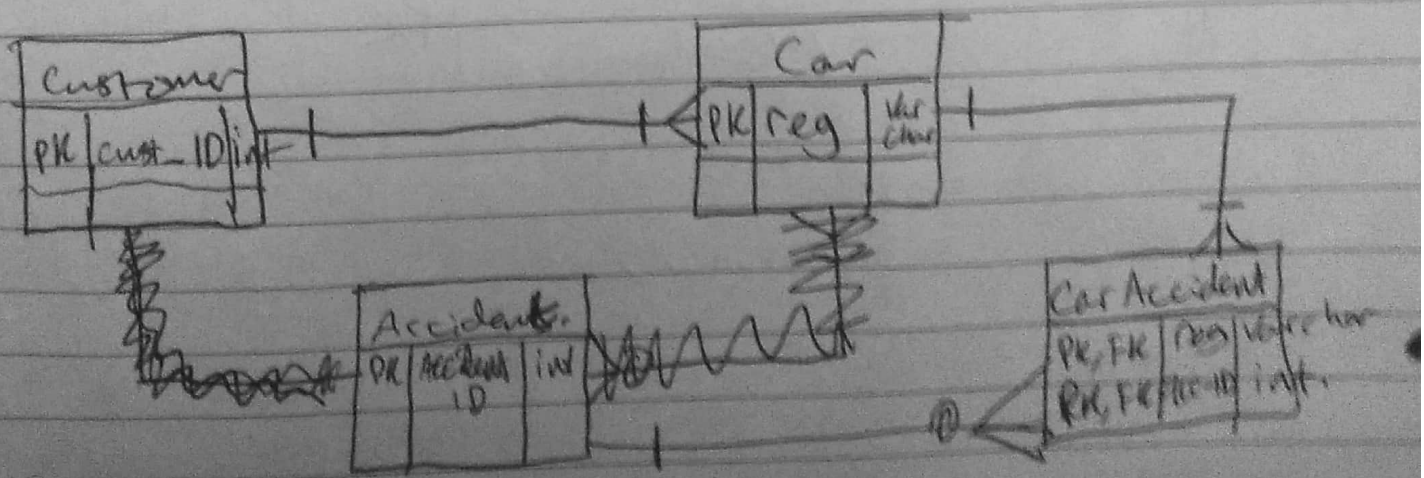
Car Insurance Company

Insured.
Customers, Cars, Accidents
Own
1:M

tables
Customers (cust_ID, CustName, cust-address

InsuredCars (incar_ID

Customer 1 ... 1:M cars
car 1:M ... 0:M accidents.



Scanned by CamScanner

Exercise 3

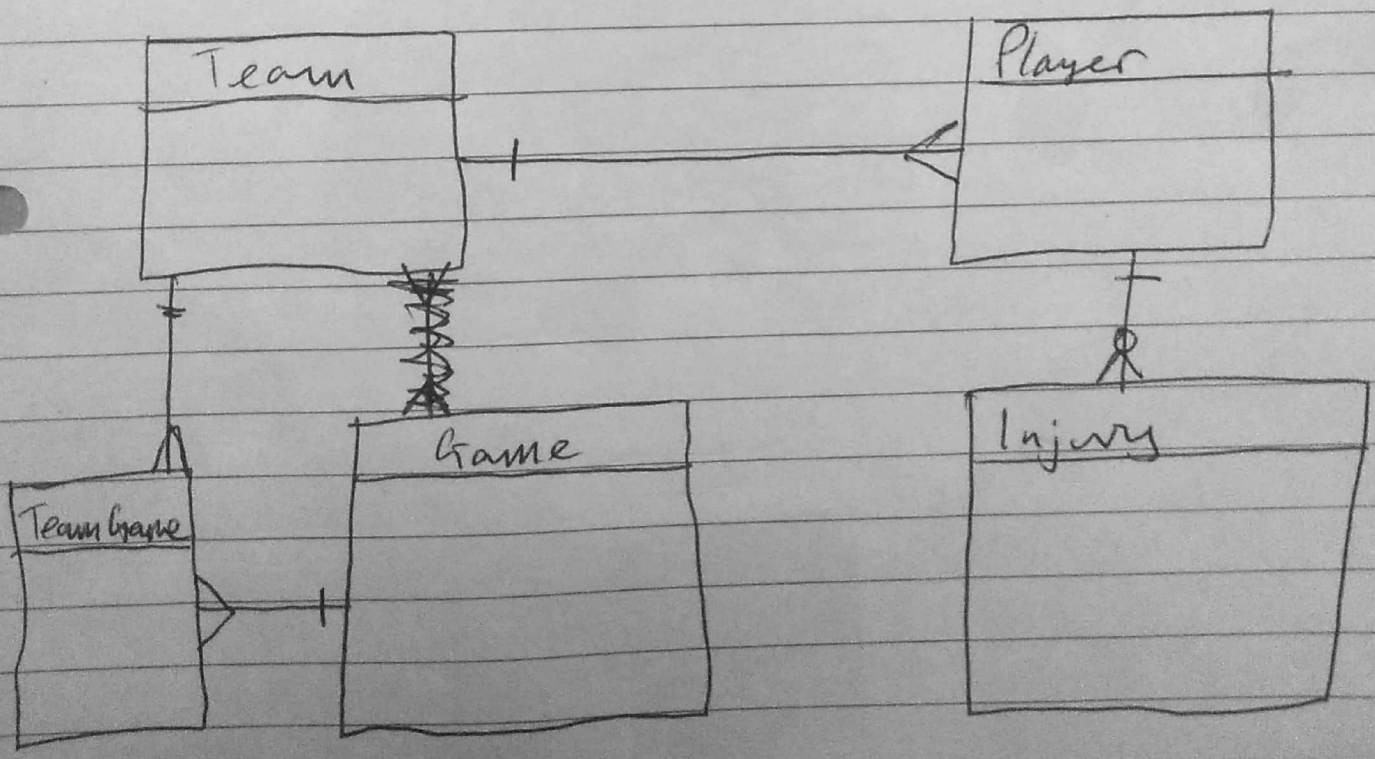
Teams (team_ID, team_city, team_Coach, teamCaptain)

Player (players_ID, player_name, player_position, player_skillLevel, player_injuries, team_ID (FK))

Games (game_ID, host_team, guest_team, game_date, game_score).

Injury (injury_ID, Player_ID)

team 1 ... m player
player 1 ... 0:m injuries
game m ... m team
~~Injury~~ ~~player~~



Before Next week do Ex 4, 5.

Scanned by CamScanner

WK 11

~~[illegible]~~ NDADD - Normalisation
↳ Making sensible tables.

EY +P CW - other entities ?
- data level.

↳ One copy of data.
↳ ~~other~~ tables that need data from another table need to "look it up".

Normalisation.
ERD ⟶ SQL

db schema - Underline PK

Attributes can be split up by whether they have repeated values.

↳ look at whether attributes are dependent on the PK.
↳ If not, make a new table and link w/ a FK

ONF - Normal form - Raw data

1NF - Intersection of each row and column contains one and only one value.
id / PK - ↳ 1 cell → one piece of data.

2NF - Attributes must depend on PK

3NF - No transitive dependancy.

Partial dependancy - Attribute dependant on part of a composite PK

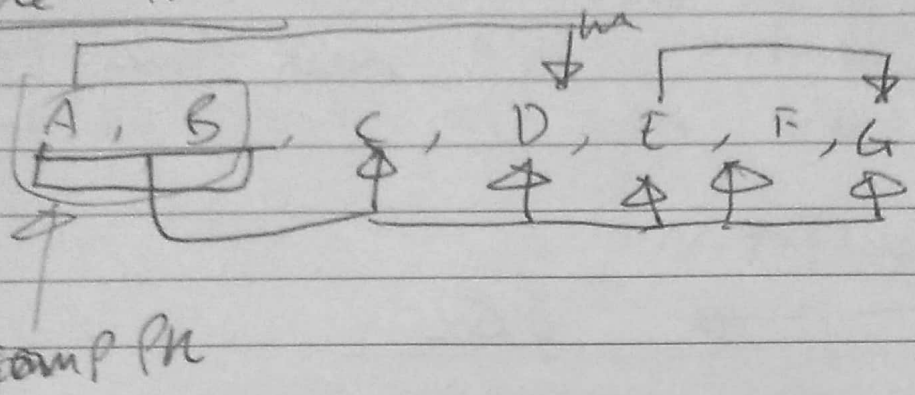
Transitive dependancy. - Attribute dependant on another attribute than the PK

Scanned by CamScanner

dependancy diagram

↳ What can you determine from the pk?

Table one



3NF

~~table one~~ tableOne (A, B, C, F)

tableTwo (A, D)

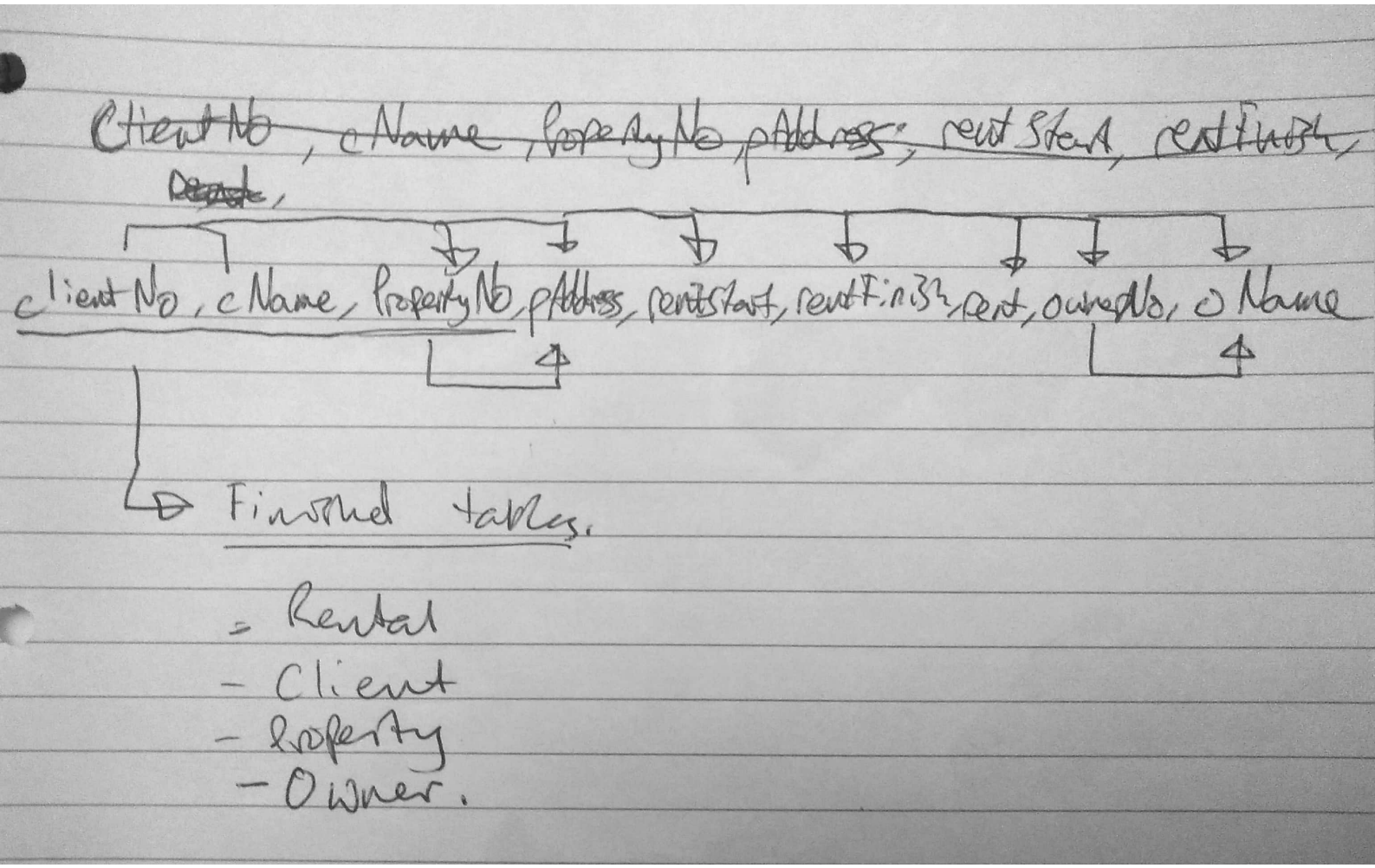
tableThree (E, G)

1NF — tableOne (A, B, C, D, E, F, G)

2NF — tableOne (A, B, C, E, F, G)

tableTwo (A, D)

Scanned by CamScanner



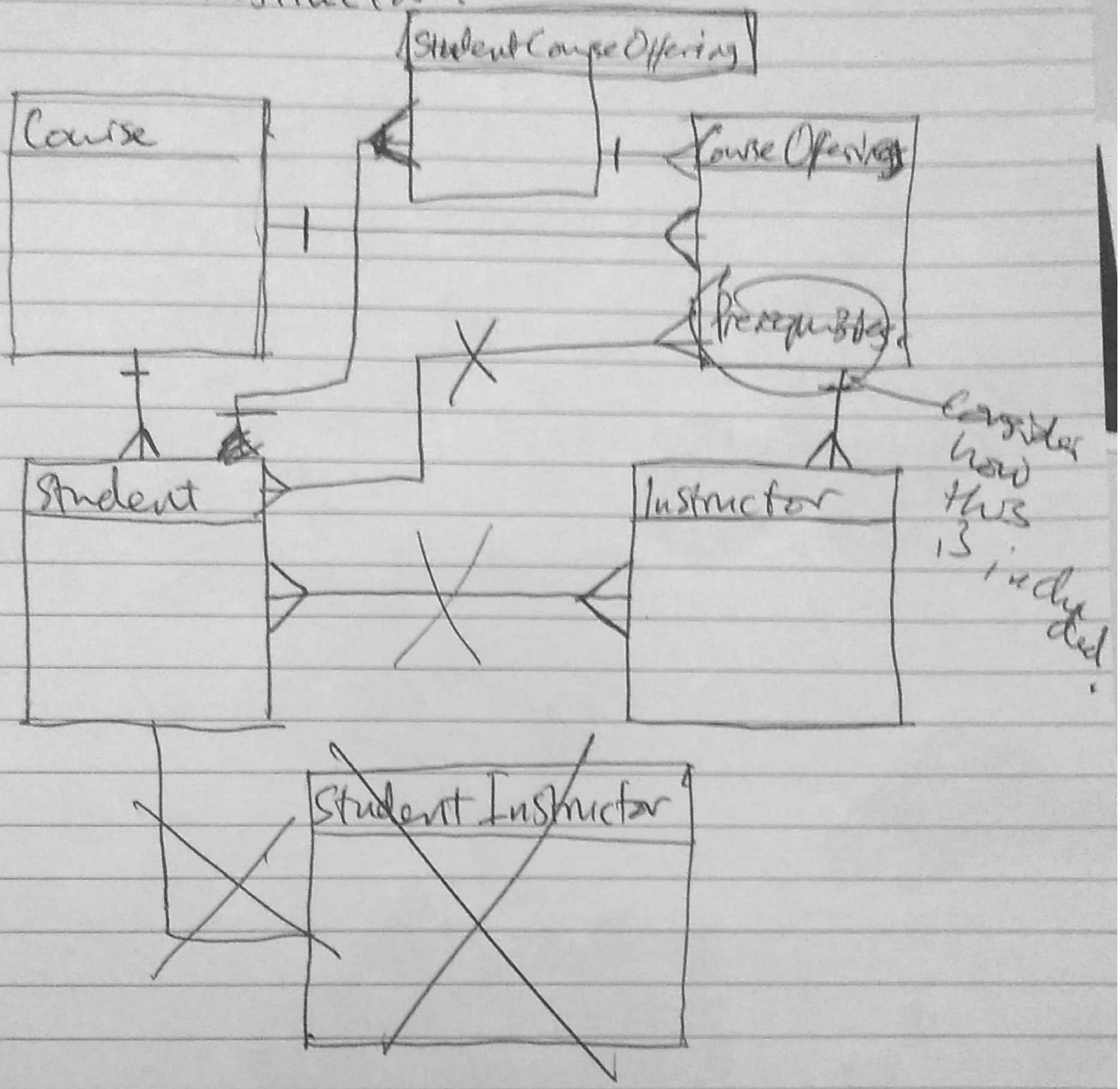
→ Finished tables.

- Rental
- Client
- Property
- Owner.

Scanned by CamScanner

INDADD PRACT W#7

- entities: course, course offering, Student, instructor.



Assumptions: Many instructors are part of one course offering.

DFD Assumptions:

Int Controls:

Scanned by CamScanner

Definitions - Reminder

- 0NF: A table that contains one or more repeating groups
- 1NF: A relation(table) in which the intersection of each row and column contains one and only one value
- 2NF: A relation(table) that is in 1NF and every primary key attribute is fully functionally dependant on the primary key (no-partial dependencies)
- 3NF: A relation(table) that is in 2NF and in which no non primary key attribute is transitively dependant on the primary key (no transitive dependencies)
- Partial dependency - an attribute dependant on only part of the primary key
- Transitive dependency - an attribute dependant on another attribute which is not the primary key

Connolly & Begg, Database Systems

Scanned by CamScanner

TOY(toyName, toyCat, ToyCatDescr, toyCost, MatCat, MatDescr, ageGrp, lowAge, highAge, eltName, eltFee)

Exercise

Convert to 2NF and then 3NF

2NF

TOY(toyName, MatCat, toyCat, toyCatDescr, toyCost, AgeGrp, lowAge, highAge, eltName, eltFee)

MAT(MatCat, MatDescr)

3NF

TOY(toyName, matCat, toyCost, toyCat, AgeGrp, eltName, eltFee)

MAT(matCat, MatDescr)

TOYCAT(toyCat, toyCatDescr)

AGE(ageGrp, lowAge, highAge)

ELF(eltName, eltFee)

Exercise

Identify the relationships and draw the ERD

14/12/2017

4

Scanned by CamScanner

WK 12

MDAPD : Normalisation part 2

- ↳ evaluating and correcting table structures to minimise data redundancies and anomalies.

Work through a series of forms → 0NF → 1NF → 2NF → 3NF

Required field data → comes from client interviews and documentation.

- 0NF → 1NF : Singular entries.
  1NF → 2NF : Remove partial dependancies.

2NF → 3NF : Remove transitive dependancies.

Databases offer a scalable solution to traditional filesystems.

- 1NF (toyName, toyCat, toyDesc, toyCost, matCat, matDesc, ageFrom, lowAge, highAge, dftName, dftFee)

Scanned by CamScanner

Wk 13

INPADP

Database Creation

L/O - data types
- SQL table creation
- ERD relevancy.

Focus = DDL - Data def lang
Later = DML - Data Manipulation Lang.

Process - create empty db
- Create tables
- insert records.

} Lifted from an ERD.
∴ ergo: no decision making.

Business rules dictate relationships.

SQL commands

```
create database <db name>
use <db name>
```

table creation:

Structure 1

```
create table TableName(
    pkName datatype [auto_increment] primary key,
    ...,
    ...);
```

Primary key can also be named within the constraints - see Structure 2.

Scanned by CamScanner

Char - fixed length string
Varchar - variable length string.

Main difference between datatypes regards storage and memory.

DATE data type : 1000! 01: 01
to
9999: 12: 31

Inserting records

insert into TableName values(...);

Remember :- comm.t command will save a db.
- Not NULL for unknown values.

SQL cribsheet! - do it.

Maintain a consistant implementation styling.

Convention: - Table names start with capitals
- attribute names start with lower.

Scanned by CamScanner

```
Create table Student (
    StudentID      INT auto-increment primary key,
    StudentFName   VARCHAR(50) not null,
    StudentLName   VARCHAR(50) not null,
    StudentDOB     DATE not null,
    Constraint fkStudent foreign key (courseID),
    courseID varchar(10) Not null,
     references Course (courseID)
);
```

```
Create table Pet (
    petID      INT  auto_increment  primary key,
    petName    VARCHAR(50)   NOT NULL,
    petColour  VARCHAR(20)   NOT NULL,
    petType    VARCHAR(20)   NOT NULL,
    petOwner   VARCHAR(50)   NOT NULL,

    constraint pk01 foreign key ...
```

```
insert into Car values (00001 'AXQ-1999',
    'Fiat', 'Red', OWN0059)

insert into Car values (00002 'AWK-1997', 'Volvo',
    'Orange', 'OWN0072')
```

don't store phone number as integers.

Scanned by CamScanner

Step 3: Insert

- Points to remember :
  - Row contents are entered between parentheses
  - Character and date values are entered between apostrophes
  - Numerical entries are not enclosed in apostrophes
  - Attribute entries are separated by commas
  - A value is required for each column
  - Use NULL for unknown values

Scanned by CamScanner

WK 13

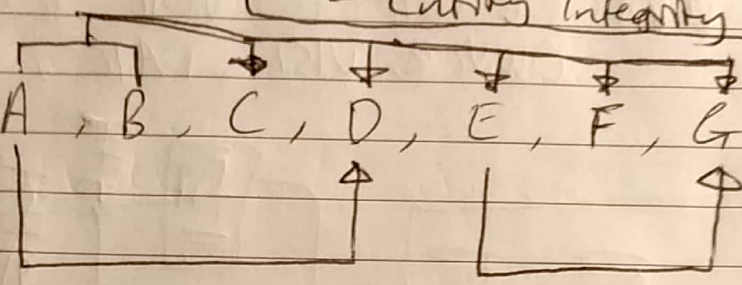
UPADD Practical Session.

Normalisation → Helps develop a stable, anomaly free database.
↳ Data integrity
- Entity Integrity, Primary key

Tantor Exercise 1

1NF

TableOne (A, B, C, D, E, F, G)



2NF

TableOne (A, B, C, E, F, G)

TableTwo (A, D)

3NF

TableOne (A, B, C, E, F)

TableTwo (A, D)

TableThree (E, G)

↳ Referential integrity
↳ Foreign keys

↳ Composite key
↳ Think about adding values for one without the other
↳ 'Cart w/o a horse'

→ Focused at the conceptual level.

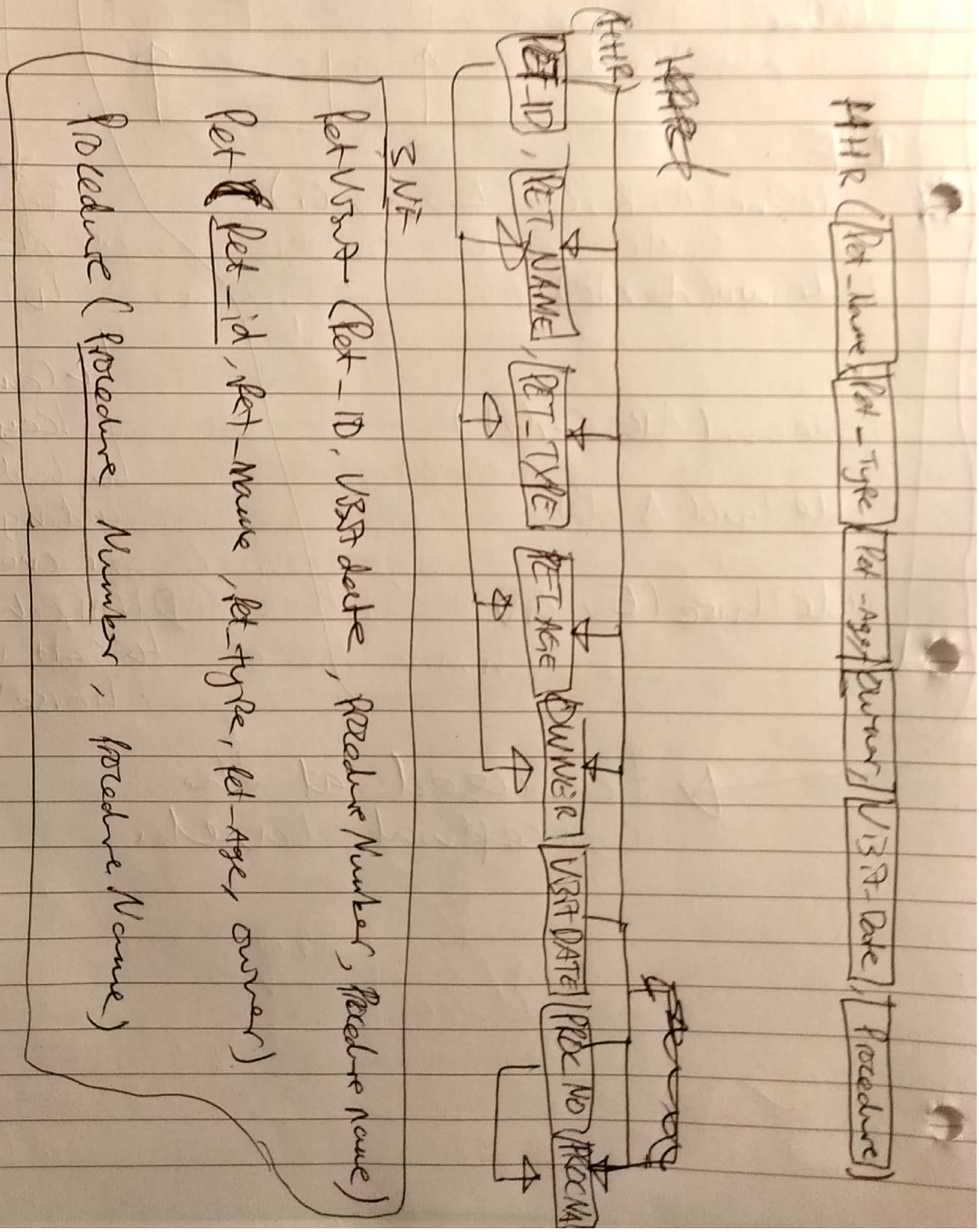
Scanned by CamScanner

#TOMATO JAPAN

Tutor Exercise 2

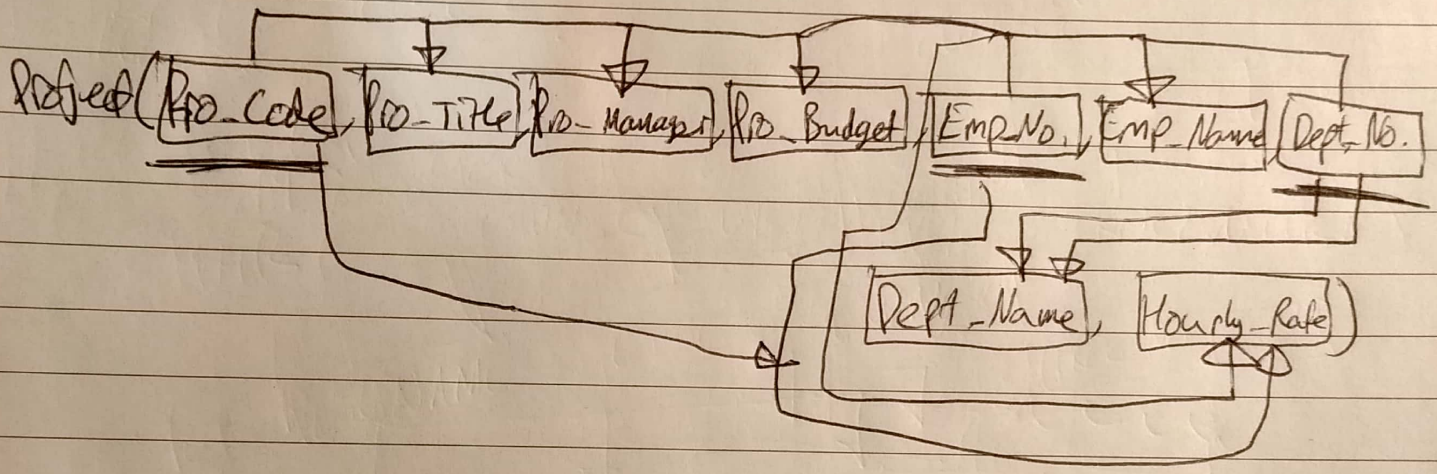
~~Problem 1~~

HEALTH-HISTORY-REPORT (PET-ID, PET-NAME, PET_TYPE, PET AGE, OWNER, VISIT_DATE, PROCEDURE)



Scanned by CamScanner

Student Exercise 1



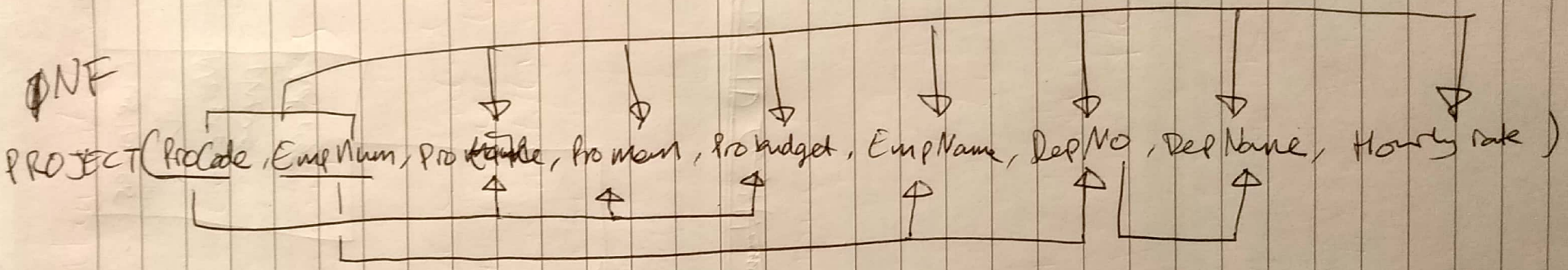
- PROJECT(Pro_Code, Pro_code, Pro_Manager, Pro_Budget, Dept_No.)

~~EMPLOYEE(EMP_No.~~

Employee(EMP_No., EMP_Name

Scanned by CamScanner

~~PROJECT (ProjectCode , Employee Number , Project Manager~~



2NF

PROJECT (ProCode, EmpNum, DeptName, Hourly rate.)

Emp (EmpNum, EmpName, Dept No)

ProJect (ProCode, ProTitle, Pro budget)

3NF

Project (ProCode, EmpNum, Hourly rate)

EMP (EmpNum, EmpName, DeptNo, DeptName)

ProJect (ProCode, ProTitle, Pro budget)

~~Dept (ProCode~~

UNDADD VK13

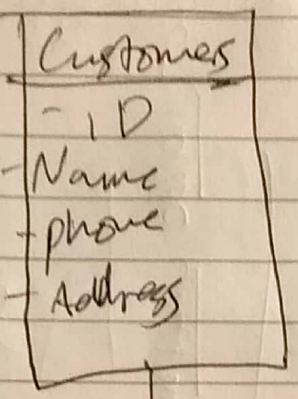
Student Exercise

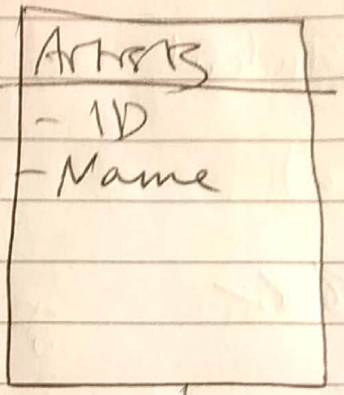
1) a)

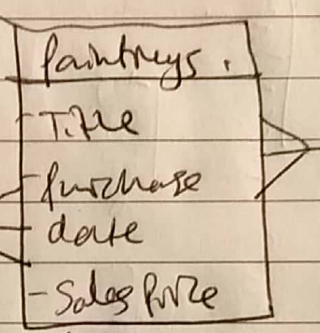
~~PROJECT~~

Scanned by CamScanner

3) a)







~~Gallery(Cust ID, Cust Name, Cust Phone, Cust Add,~~

Purchase:

~~Gallery~~(CustID, Cust Name, Cust Phone, Cust Add, Purchase Title, Paint Pr, Paint Price, Art ID, Art Name)

Purchase (cID, pTitle, pPurchase Date, cName, cPhone, cAdd, pPrice, aID, aName)

Purchase (cID, pTitle, pPurchase Date, pPrice)

Customer (cID, cName, cPhone, cAdd)

Artist (pTitle, aID, aName)

~~Painting~~

Primary Keys

Method 1

```
create table Student(
   stuID int auto_increment primary key,
   ...
);
```

Method 2

```
create table Student(
   stuID int auto_increment,
   ...
   primary key(stuID)
);
```

Method 3

```
create table Student(
   stuID int auto_increment,
   ...
   constraint stuPK primary key(stuID)
);
```

Scanned by CamScanner

Composite Primary Keys

Method 1

```
create table Lecturer(
   lecFName varchar(30)primary key,
   lecLName varchar(30)primary key,
   ...
);
```

Method 2

```
create table Lecturer(
   ...
   primary key(lecFName, lecLName)
);
```

Method 3

```
create table Lecturer(
   ...
   constraint lecPK primary key(lecFName, lecLName)
);
```

Scanned by CamScanner

Foreign Keys

Method 1

```
create table Cat(
    ...
  ownerID varchar(8) foreign key references Owner(ownerID)
);
```

Method 2

```
create table Cat(
    ...
  ownerID varchar(8)
  ...
  foreign key (ownerID) references Owner(ownerID)
);
```

Method 3

```
create table Cat(
  ...
  constraint foreign key (ownerID) references Owner(ownerID)
);
```


Scanned by CamScanner

WK14

INDADD: Database Manipulation

L/O - Additional data types
- Use SQL to alter tables.

3 ways to declare a primary key.

Foreign keys - reference linked table
- Attribute names do not need to match.

Forget a:- pk?
- Attribute.

How to :- Add an attribute?
- Remove an attribute?

Alter a table

1 - declare desired table
2 - add / modify / drop.

Altering a PK

1- declare
2- add / drop

ALTER TABLE Cat MODIFY CatName VARCHAR(25);

ALTER TABLE Cat ADD FOREIGN KEY(cat_breed)
REFERENCES Breeds (Cat_breed);

ALTER TABLE Cat MODIFY CatBreed VARCHAR(20);

Scanned by CamScanner

ALTER - db structure

UPDATE - data.

=

~~INSERT INTO Students (~~

ALTER TABLE Student
ADD FOREIGN KEY (courseID) REFERENCES Course(ID)
ADD FOREIGN KEY (tutorID) REFERENCES Tutors(ID);
INSERT INTO Students ('~~Assaf~~', 'UP456789', 'Assaf',
'Naille', '05243 123 456');
UPDATE Student SET StuName = "Ward" WHERE StuName
= "Smith";

EDIT STRUCTURE BEFORE DATA.

=

~~ALTER TABLE ArtIst ADD FOREIGN KEY(artID)~~
~~CONSTRAINT FKO1 FOREIGN KEY~~
~~(artID);~~

~~REFERENCES~~

ALTER table ArtIst add constraint art ~~PK~~ primary
key (artID);

ALTER table Album add constraint album ~~PK~~ primary
key (albumID);

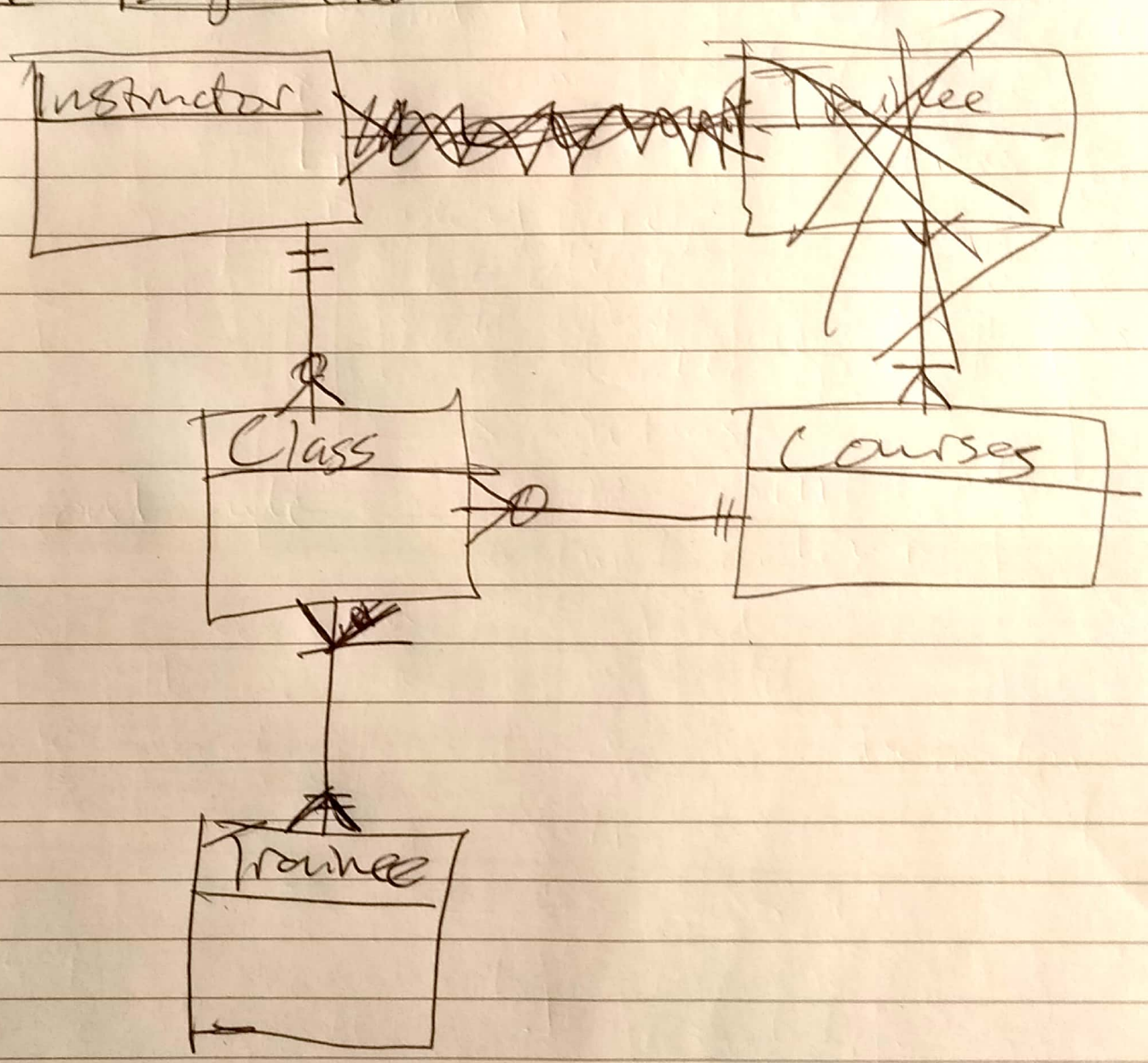
Alter table ~~Album~~
add column albumArtIst int,
add albumFKO1 foreign key (albumArtIst references
ArtIst(artID);

Scanned by CamScanner

WK 14

WDADD Practical

HEG rough ERD



Scanned by CamScanner

Wk 15
INDA00

Database Queries

* delete columns
* drop table / database.

* SQL queries are the interface between the UX and the data.

* Queries must provide information of use

* Queries need business meaning

* Profit / Loss analysis,

Dream homes - data with valuable business meaning.

- Most Popular (?) areas
  - highest turnover of houses
  - highest amount of houses concurrently being sold.
- Region where the most expensive houses sold.
  - Average value
- Expiring contracts.

Scanned by CamScanner

Conditional Selecting

Select * from <table> where <field> = <value>;

↳ or a comma separated list of desired attributes.

* Boolean operators for extra conditionals

where <attribute> = <value> and/or <attribute> = <value>;

Subexpressions in brackets operated on first

Also **wild cards** with **Like** keyword

Also : = , <= , >= , !=

Aliasing can be used when selecting attributes.

Select fNam as FirstName,

⇒ **Exercise**

Select fName as FirstName,
LName as LastName,
Staff_Salary as Salary
Where Salary = 9000 or Salary = 12000;

Scanned by CamScanner

Keyword between: for a range of values. (inclusively).

```
Select fName as FirstName,
       LName as LastName,
       Staf_salary as Salary
Where Salary between 30000 and 9000
```

Keyword: distinct
└ Eliminates duplicate data.

Keyword: limit, after 'where' conditionals, reduces list of entries to a specific number.

Comments
single line: --
multi line: /* ...
*/

Scanned by CamScanner

WK16

INDA00 More Database Queries.

Keywords: min, Max, count, avg, Sum, group by, order by, having, asc, desc, JOINS

MIN() - returns smallest value of selected column

MAX() - " largest " " "
"

```
Select min(Salary) as LowestSalary,
       max(Salary as HighestSalary
from Staff;
```

AVG - returns average of values in a column

```
Select avg(Salary) as AverageSalary
from Staff
where branchNo = "B65";
```

SUM() - returns sum of values in a column

```
Select sum(salary) as TotalSalary
from Staff
where branchNo = "B65";
```

COUNT() - returns the number of values in a selected column.

```
Select count('jobRole) as ManagerCount
from Staff
where jobRole = "Manager";
```

Scanned by CamScanner

group by

- Used in conjunction with the aggregate functions to group the result-set by one or more columns.

```
Select branchNo, count (propertyNo)
from PropertyForRent
group by branchNo;
```

having

- like a where clause but used specifically with a group query. Filters groups not individual rows.

```
Select branchNo, count (StaffNo)
from Staff
group by branchNo
having count (staffNo) > 1;
```

order by / asc / desc

- Sort results of a query with multiple attributes.

```
Select StaffNo, fName, lName, Salary
from Staff
order by salary desc;
```

Scanned by CamScanner

Into

- Copies data from one table into another.

```
Select *
into OwnersBackup
from owners;
```

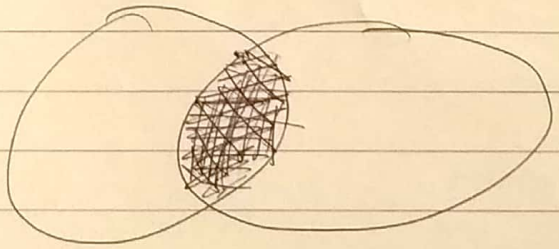
Simple PK/FK Join

- Produce information from multiple tables

```
Select *
from PrivateOwner, PropertyForRent
where PrivateOwner.OwnerNo = PropertyForRent.ownerNo;
```

Inner Join

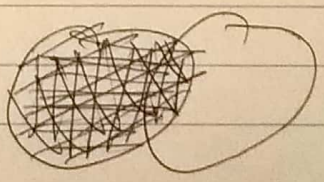
- Produce set of results matching both tables



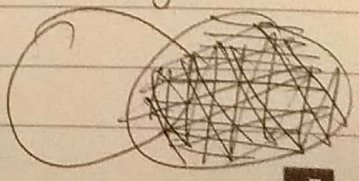
INNER JOIN

LEFT/RIGHT JOIN

Left



Right



Scanned by CamScanner

WK(7

INDADD dB Queries 3

Databases get their use from the questions that can be asked of the data.

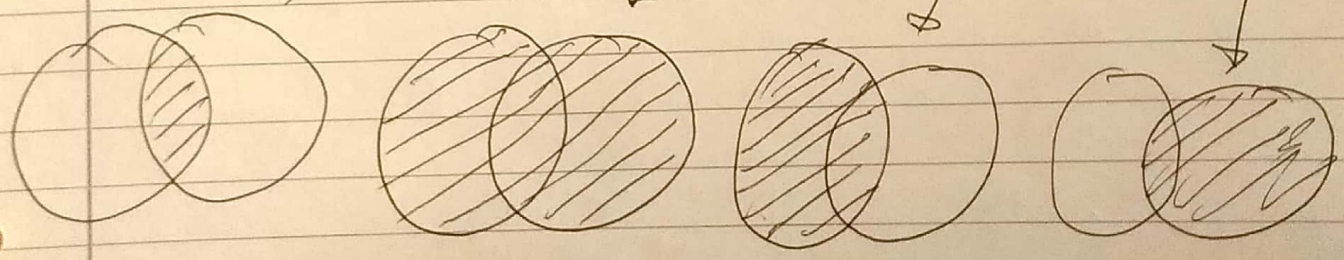
What are the questions that need answering? (Letting agent)

- Cities with the highest turnover of ~~[illegible]~~ renters.
  - Average rent in a city.
    mean, modal, median?
    - Average number of rooms.
      - Average rent / room for each city.
        - List of properties in a city between a rent range
          - Search Attorney properties.

⇒ No joins in the exam.

Joins

Inner join, outer join, Left join, right join.



CORPORATE CONDITIONING

Scanned by CamScanner

wk 18

INDADD - More SQL

String Functions

- length()
  - how long is a string?
- Concat()
  - Concatenate 2+ strings.
- concat_WS()
  - concatenates 2+ strings with a separator.
- lower(), upper(), reverse()
- substring → part of a string

substring("Shelley", 3, 5)
- 3 → start
- 5 → number

⇩

"elley"

Numeric Functions

- abs(), ceiling(), floor(), round()
- power(), pi(), sqrt().

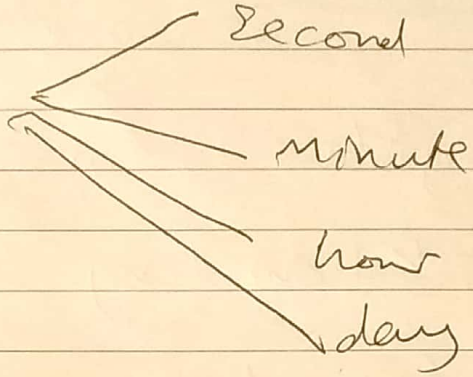
Scanned by CamScanner

- curdate()
- curtime()

- date()

- datediff()

- week()
- month()
- year
- last_day()



- "What are the applications for these functions?"

- Calculations can be output as a singular field.
  - ↳ can also be done in table creation

SELECT price + shipping AS total_price FROM products;

Nested Queries

- ↳ ~~nested~~ analysis
  - ↳ "Whose salary is greater than average?"
- ↳ Joins can make things easier rather than trying further nesting.

Scanned by CamScanner

Subqueries vs Joins

- Joins provide a performance boost.
  - Joins > Subqueries
  - Subqueries provide a good fall back.

Reading Week next wk.

- Progress so far.
- Start thinking about SQL CW.

Scanned by CamScanner

WK10

Recovering content

INDADD

Rel. Abs
↳ model real life systems
↳ Information → /breakdown/ → Data

DDL → changes the Schema.

DML → Changes the data

INT / SMALLINT - no number after.

BLOB - binary large object, storing files.

Joining Tables
↳ helps make information out of data -

Join joinName on Table1.att = Table2.att where;
↳ Can have multiple joins for degrees of separations
↳ No limit.

Mockaroo

Show warnings → data truncation.

CW Queries : - 5 ~~of them~~ basic queries
- Meaningful
- 5 complex queries → Multiple tables, joins

Scanned by CamScanner

- Semi-Structured data
  - an idea of how data could be structured
- Un-structured data
  - Without an idea of how the data would be stored.

- Relational Algebra.
- Optimisation queries.

Why / What if ?

---

* One event can have ~~many~~ one to many suppliers, One supplier can ~~Supply~~ one to many events

* ~~One can be sponsored to zero to many Event~~

* One event can be sponsored by zero to many sponsors, one sponsor can sponsor zero to many events.

Scanned by CamScanner

WK21

INDADD : Security

Data security

↳ Data often the most valuable corporate resource
↳ vulnerable to theft, loss and damage

↳ Mechanisms protecting database against intentional / accidental threats.
↳ internal / external.
↳ disgruntled employees.

dB Environment

↳ Security must cover:
- The database
- The web server
- The Internet
- The Client's machine

} Who is responsible for these?

CIA Triangle

↳ Confidentiality, Integrity, Availability.

Main Issues

↳ Threats - potential for harm/loss

↳ Vulnerabilities - Potential for a threat

↳ attack - an executed threat

↳ System failure - Total successful execution of a threat.

Scanned by CamScanner

Threat / Failure Types

- System crashes, main memory loss
- Media failure; loss of secondary storage
- Application software errors
  - difficult to detect → system can continue as normal
- Physical/Natural disasters
- Carelessness - lack of training
- Regular 'ol sabotage.

Threat Impact

- number of factors?
  - what has been the result of an attack.
    - Hardware / software condition?
    - Backup condition?
    - Restoration time?
    - Data recovery?

Counter Measures

- Computer
- Non-computer

Scanned by CamScanner

Computer Counter Measures

- Authentication → passwords.
  - Authorisation
    - What access is granted with a certain login.
      - Database level with SQL
        - keywords authorised for certain users.
          - privileges stored in the system catalogue.
            - Set up classes.
- Views
  - Dynamic Result
  - Virtual tables
  - What you see, depends on who you are.
- Back Ups
  - Periodically copying data
    - Log Files
      - Contains information about updates to the database
        - Transaction policy
    - policy → back up types and Frequency.
      - Full, Differential, Incremental, continuous.

Scanned by CamScanner

- Computer center measures
  - Integrity → prevents unauthorized.
    - Domain Constraint
    - Entity Integrity
    - Referential Integrity
    - General Constraints.
  - Encryption
    - Data obfuscation, decrypted with a key.
  - RAID - Redundant Array Independent Disks
    - Instantaneous replacement if hardware is compromised.

Non Computer Based Controls

- Policies
  - Contingency plans
  - Personnel controls
  - Secure operational procedures
  - Maintenance Agreements
- Physical Access Controls.

Scanned by CamScanner

Security Policy

- Requires Appropriate Controls
  - security policy statement
- Cost effectiveness / usability.

Scanned by CamScanner

WK22

LAST LECTURE!

INDADD : Transaction Management and Concurrency Control

Covers :- Locking Methods
- Deadlocks.

Transaction?

- Action that reads from and/or writes to a database.
  - Select, update, insert.
- Eg - Charging a purchase to a customer account
  - write invoice
  - reduce inventory quantity
  - update account transactions
  - Update customer balance.
- A logical unit
  - entirely completed or aborted.
- Success: changes a database from one constant state to another.

Evaluating Transaction Results

- improper / incomplete transaction can severly effect integrity.
  - DBMS helps with this

Rollback - Undo transactions.
- Can happen in the middle of a transaction

Transactions end at a COMMIT

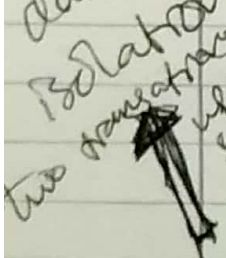
Transaction Log

- record for the beginning of transaction
  - for each transaction component.
    - operations
    - object names
    - before and after values
    - pointers to previous and next transactions.

Concurrency Control

- Coordination of simultaneous transaction execution in a multiprocessing database system
  - Ensure serializability of transactions in a multi user database environment.
  - can create integrity and consistency problems.
    - lost updates
    - Uncommitted data
    - Inconsistent retrievals.

data isolation: two transactions do not update the same data.



Scheduler → efficient CPU use

- DBMS program
  - Establish order of query execution
  - Concurrency control algorithms.

Scanned by CamScanner

T1	T2	Conflict?
R	R	N
R	W	Y
W	R	Y
W	W	Y

Lock Granularity → What level are things locked?

- Database-level
- Table-level
- Page-level
- Row-level.
- Field-level

Deadlocks

- Two transactions waiting for e/o to finish
- control through:
  - Prevention
  - Detection
  - Avoidance.

Scanned by CamScanner