# 1. Basic knowledge

a. relational algebra

π select

σ where

⋈ join

b. update, delete, insert anomaly

Insert Anomaly

–Insert a new employee only if they are assigned to a project

Delete Anomaly

–Delete the only employee assigned to a project?

–Delete the only employee of a particular job class?

Modification (or update) Anomaly

–Update a job class hourly rate - need to update multiple rows

c. strong(identifying) relationship, weak(non-identifying) relationship

Strong relationship:

- Child entity is existence-dependent on parent
- PK of Child Entity contains PK component of Parent Entity
- Usually occurs utilizing a composite key for primary key, which means one of this composite key components must be the primary key of the parent entity.

Weak relationship:

- Entity is existence-independent of other entities
- PK of Child doesn’t contain PK component of Parent Entity

d. strong entity, weak entity

Strong Entity:

- has a key which may be defined without reference to other entities

Weak entity:

- has a key which requires the existence of one or more other entities.

e. all kinds of key(primary, foreign, candidate, super)

PrimaryKey:

A primary key must be chosen considering the data that may be added to the table in the future

ForeignKey:

An attribute/s in a relation that exists in the same, or another relation as a Primary Key.

Candidate key

- CK of a relation R is an attribute or set of attributes which exhibits the following properties: – Uniqueness property (as above), and – No proper subset of CK has the uniqueness property (Minimality or Irreducibility property) ie. a minimal superkey

Super key

- A superkey of a relation R is an attribute or set of attributes which exhibits only the uniqueness property

f. simple attribute, composite attribute

Simple

– Cannot be subdivided

– Age, sex, marital status
▪ Composite
– Can be subdivided into additional attributes
– Address into street, city, zip

g. single-valued attribute, multi-valued attribute

▪ Single-valued
– Can have only a single value
– Person has one social security number
▪ Multi-valued
– Can have many values
– Person may have several college degrees

h. data integrity

Entity integrity
– Primary key value must not be NULL.
• No duplicate tuple property then ensures that each primary key must be unique
▪ Referential integrity
– The values of FK must either match a value of a full PK in the related relation or be NULL.
▪ Column/Domain integrity
– All values in a given column must come from the same domain (the same data type and range).

## 2. Conceptual/Logical model

a. 1:M relationship
b. insert bridge entity to M:M relationship
c. multivalued attribute needs to create an entity
d. more than 2 relationships between two entities
e. relationship within one entity
f. 1:1 relationship
g. KEEP IN MIND! assumptions or no assumption is needed (For exam purposes)

## 3. Normorlization

repeating group, partial, transitive dependency
careful

## 1. SQL

a. select...where…(don’t forget order by)
b. medium difficult aggregate function

AVG, COUNT, MIN, MAX, SUM

c. DDL

do not add commit at the end!

d. DML

(delete,update,insert-->commit)

e. oracle functions (NVL, LIKE, extract, to_char, to_date, ceil, floor, round, initcap...)

NVL(xxx, ‘N’) - replace null value with ‘N’

LIKE ‘m%’ 代表多个字符 ‘%m%’, ‘%m’

‘m_’ 代表单个字符

区分大小写

extract - select extract (year from sysdate) from dual;

ceil - round up 9.3--10

floor - round down 9.7--9

round - set decimal digit

initcap - capitalize

f. hard subquery (multiple columns), subquery (correlated), subquery (inline)

g. sequence

create sequence student_seq start with 100 increment by 1;

h. case

```
case
        when … then …
        when … then ...
end
```

i. left/right join

j. union, union all, intersect, minus

union - combine two ‘select tables’ only if data types and the number of columns are the same (delete repeated values automatically)

AUB

union all - same as union except without deleting repeated values

AUB+A∩B

intersect - gain intersect from two tables (attributes appear in both tables)

A∩B

minus - AUB-B

k. join itself

## 2. NoSQL

a. characters of big data

VVV:

**Volume: The quantity of data to be stored**

**Variety:  Variations in the structure of the data to be stored.**

**Velocity: The speed at which data enters the system and must be processed.**

Variability – Data meaning changes depending on context

Veracity – Correctness of the data

Value – Data can provide meaningful information

Visualisation – Data can be presented in a way which can be easily understood

scaling up: keeping the same number of systems but migrating each one to a larger system

For example buying a new server with larger memory, bigger storage space and faster CPU.

scaling out: when the workload exceeds server capacity, it is spread out across a number of servers

   For example buying many common computers and using distributed systems for storage and processing.

b. comprehension and transaction of JSON form

   select json_object (‘xxx’ value xxxx,
   ‘yyy’ value yyy format json) || ‘,’
   from xxx
   where xxx
   group by xxx
   order by xxx;

c. Hadoop

   fuck you

d. Map Reduce -

   Framework used to process large data sets across clusters.
   Breakdown complex tasks → filters into a set of key-value pairs → reduce results and produce a single result

e. Mongo DB (CRUD)

   create
   db.collection.insertOne(...JSON...);
   db.collection.insertMany([JSON1,JSON2,...]);
   retrieve
   db.collection.find({},{‘_id’:0, “name”:1});
   db.collection.find({}).pretty();
   db.collection.find({“address”:/.*Melbourne.*/}).count()
   db.collection.find({$and:[{“age”:{$gt:18}},{“age”:{$lt:28}]}});
   update
   db.collection.updateOne({},{$set:{“xxx”:123}});
   db.collection.updateOne({},{$set:{“xxx.$.yyy”:456}});
   db.collection.update({},{$push:{}}); --add data from array
   db.collection.update({},{$pull:{}}); --remove data from array
   delete
   db.collection.deleteOne({});
   db.collection.deleteMany({});

## 3. Transaction management

a. ACID

   Atomicity:
      all database operations (SQL requests) of a transaction must be entirely completed or entirely aborted.
   Consistency:
      it must take the database from one consistent state to another.
   Isolation:
      it must not interfere with other concurrent transactions
      data used during execution of a transaction cannot be used by a second transaction until the first one is completed.
   Durability:
      once completed the changes the transaction made to the data are durable, even in the event of system failure.

b. shared lock, exclusive lock

c. lock table
d. Deadlock
e. transaction log
f. How to prevent a deadlock?

   prevent:
   A transaction must acquire all the locks it requires before it updates any record.
   If it cannot acquire a necessary lock, it releases all locks, and tries again later.

   recovery:
   Resolution involves having the Lock Manager force one of the transactions to abort, thus releasing all its locks.
g. Recovery and Restart
   - Restart – Soft crashes • loss of volatile storage, but no damage to disks. These necessitate restart facilities.
   - Recovery – Hard crashes • anything that makes the disk permanently unreadable. These necessitate recovery facilities.