most 'Tables' you work with in an organization are actually views

When you create a view you can prefix it with VIEW_ or even VW_

For tables you can prefix it TBL_

You can create privileges to restrict what users can see

DESIGNING DATABASES

apply data modelling tech to DB design

normalize

restructuring data to
defined normal forms

eliminates data redundancy
+ inconsistencies

3 common forms

FIRST NORMAL FORM 1NF

2NF

3NF

1NF

EACH FIELD IN A TABLE
SHOULD CONTAIN A
SINGLE VALUE

EACH ROW IS UNIQUE

row can have a
repeat fa
But whole row don't match

family	children
Smiths	Chris Abby Susan
Jones	Steve Mary Dillon

family

Smiths	Chris
Smiths	Abby
Smiths	Susan
Jones	Steve
Jones	Mary
Jones	Dillon

2NF

BE IN 1NF

SINGLE COLUMN PK

PK

IDENTIFIES ROWS UNIQUELY

GENERALLY THERE COULD BE A NEED TO CREATE A NEW TABLE

Smiths A
Smiths
Smiths
Jones
Jones
Jones

familyTable

1	Smith
2	Jones

Child Table

ChildID	familyID	Children
11	1	
22	1	
33	1	
44	2	
55	2	
66	2	

TRANSITIVE DEPENDANCE IS THE RELIANCE OF A COLUMNS VALUE ON ANOTHER COLUMN THROUGH A THIRD COLUMN

TRANSITIVE

IF X > Y AND Y > Z
THEN X > Z

DEPENDANCY

ONE VALUE RELIES ON ANOTHER
CITY RELIES ON ZIP
AGE DEPENDS ON BIRTHDATE

say you have 3 cols
StoreName OwnerName OwnerAdd

OwnerName + OwnerAdd DEPENDON StoreName

OwnerAdd also relies on the OwnerName

so OwnerAdd relies on StoreName via OwnerName

3NF

IN 2NF

CONTAIN NON TRANSITIVELY DEPENDAND COLUMNS

PET NORMALIZER

No one table should exist in isolation from all other tables

always have a relationship

FOREIGN KEYS

always make a table have a primary key.

This is how you "talk to it" - PRIMARY KEYS ARE UNIQUE

eg in pets

owner has ID PK

pet has ID PK

foreign key references key in another table

REFERENTIAL INTEGRITY.

we can tie table together using FK

(FK does not have to reference PK in other table)
But it must be UNIQUE

FOREIGN KEY (columnname)
REFERENCES
table-name (colname)

If there is a FK and you insert values then the fk has to exist in the table the FK refers to

You will get an error that says violates FK constraints

Key (colname) = (val) is not present in table table name

Because of Referential Integrity

It will not let you add if it does not exist.

That way you eliminate REDUNDANCY.

eg no child will exist without a parent
no pet without an owner

Also it will not let you UPDATE things ~~unless~~ because they exist

You cant delete the parent you have to remove the child
- you have to delete them in order

THAT WAY YOU MAINTAIN DATA INTEGRITY (REFERENTIAL INTEGRITY)

DATA MODELING

Foreign keys reference the PK of another table
can have a diff name
need not be unique

Relationships
one To one
one To many
many To many

1-1 ONE TO ONE
each item in one col is linked to only one item from another col

EG SSN — PERSON
1 1

ONE TO MANY

ONE ADDRESS TO MANY PPL
(each person has an address

So if some family moves you just have to change the address once in the master address table

MANY TO MANY

ID	Child
1	Bart
2	Lisa
3	Maggie

ID	Parents
11	Homer
12	Marge

EACH CHILD CAN HAVE MORE THAN ONE PARENT

EACH PARENT CAN HAVE MORE THAN ONE CHILD

In a many to many relationship

The 2 Tables are joined in a JUNCTION TABLE

Junction Table

parent_id	child_id
11	1
11	2
11	3
12	1
12	2
12	3

Join the child and parent table to junction table

Homer	Bart
Homer	Lisa
Homer	Maggie
Marge	Bart
Marge	Lisa
Marge	Maggie

```
CREATE TABLE child_parent
  child_id INTEGER NOT NULL
  FOREIGN KEY (child_id) REFERENCES
  children (child_id
  parent_id INTEGER NOT NULL
  FOREIGN KEY (parent_id) REFERENCES
  parents (parent_id)
  PRIMARY KEY (child_id, parent_id)
```

JUNCTION TABLE

```
SELECT children.child_name,
       child-parent.child_id,
       parents.parent_name,
       parents.parent_id

FROM children
LEFT JOIN child-parent
ON
  child-parent.child_id =
   children.child_id
LEFT JOIN parents
ON
   child-parent_parent_id =
    parent.parent_id;
```

INNER JOIN
When you join you drop records that don't match

OUTER JOIN
Matches everything

Left
keeps integrity of leftmost table
anything that does not match gets dropped

VENN DIAGRAM

ENTITY RELATIONSHIP DIAGRAM

a visual representation of entity relationships within a database

How are entities related

primary key
foreign key
one to many
one to one
many to many

Before you create the DB (tables) you map out all the relationships in a diagram

Conceptually draw a diagram

Iterate and improve on diagram that represents your DB before you create it

You can also create a view diagram to show all relationships

Entities
Their data types
and relationships
are all illustrated in ERD

There are 3 models used
when creating ERD

Conceptual
Basic info containing
table names + col

Logical
Slightly more complex
with IDs and data types

physical
The BLUEPRINT of your DB

REFLECTS PHYSICAL
RELATIONSHIPS BETWEEN
ENTITIES

Gym ex
conce
logica

→ if yo
so lin

→ relate

→ link

So
Train
of

you c
m
refe

In qu

Pay

Gym example

conceptual.txt

logical.txt

→ if you dont pay you cant go so link member to payments

→ relate trainer to members

→ link trainer to a gym

So create a FK in Trainer that links to PK of Gym

you cant make payment if not Member may add FK member ID reference payment PK

In quickdatabase diagram

PaymentId INT FK-Payments.ID

— is for one to one

look under docs to get a full list

UNION (optional)

one liner

complete list from 2 tables

Toys

UNION

```
SELECT toy_id AS
  id, type
FROM toys
UNION
SELECT game_id AS id, type
FROM games
```

If you want to include
duplicates you say
UNION ALL
Just operates a little bit faster

next adv data storage + retrieval