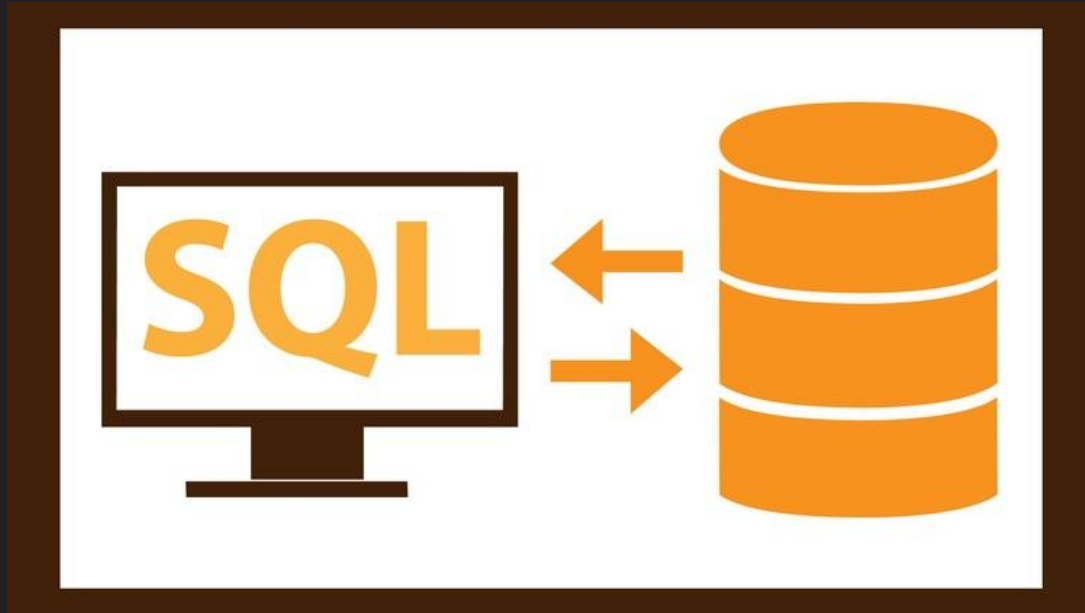


# Database Management for Pulsar Data- A fight against entropy



# The Intro

*When dealing vast amounts of data,  
organizing/bookkeeping solves half the issues*

-

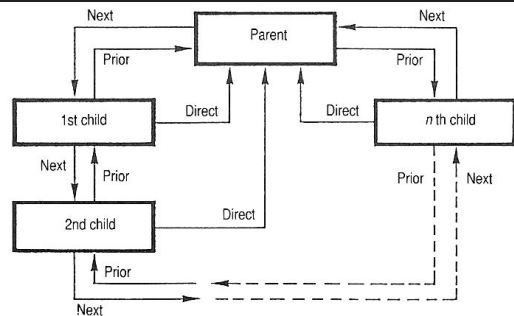
Database management systems are a reliable resource.

# Why do we need databases?

- To manage large chunks of data:
- Accuracy: When doing data entry files in a spreadsheet, it becomes difficult to manage the accuracy as there are no validations present in it.
- Ease of updating data: With the database, you can flexibly update the data according to your convenience. **Moreover, multiple people can also edit data at same time!!.**
- Security of data: You have security groups and privileges you set to restrict access.
- Data integrity: You can be assured of accuracy and consistency of data due to the built in integrity checks and access controls.

# A bit of history...

## Charles Bachman



A closed chain of records in a navigational database model (e.g. CODASYL), with **next pointers**, **prior pointers** and **direct pointers** provided by keys in the various records.

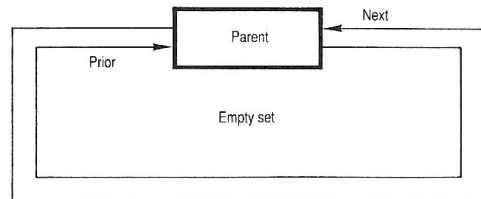


Illustration of an **empty set**

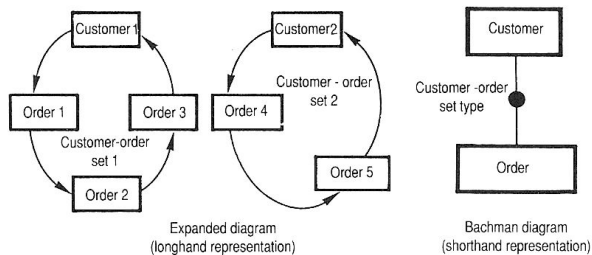


Illustration of a set type using a **Bachman diagram**

The record set, basic structure of navigational (e.g. CODASYL) database model. A set consists of one parent record (also called "the owner"), and  $n$  child records (also called members records)

# Types of databases

## Relational DBMS

- Tables and rows based
- SQL used to talk to db

**Adv:** Easily readable and manageable

**Disadv. :** Vertical scaling expensive

## Non-Relational DBMS

- Document format; optimized for data types
- NoSQL used to talk to db

**Adv:** scalable

**Disadv. :** Not well understood

# SQL : Structured Query Language

- Query describes result but not how to do it
- Has a set of procedures to deal with querying in an optimised way.
- Scalability resolving: Sharding or NoSQL

Some unique and useful features:

- a) Indexing
- b) Referential integrity
- c) Unique Constraints



# Back to relevance for Pulsars!

Large Surveys



More data



Lots of bookkeeping!

Where to start?

# Schema

- A blueprint to give the relevant category objects with reference to which data can be stored.
- This is where you are spoilt for choice and can plan out everything!



## Beamformer\_Configuration

bf_config_id	@* int(11)
centre_frequency	float(10)?
bandwidth	float(10)?
nchans	int(10)?
tsamp	float(10)?
metadata	text?

## Targets

target_id	@* int(11)
project_id	int(11)
source_name	text?
ra	float(10)
dec	float(10)
region	text
semi_major_axis	float
semi_minor_axis	float
position_angle	float
metadata	text?
notes	text

## Data\_Products

dp_id	@* int(11)
pointing_id	int(11)
beam_id	int(11)
processing_id	int(11)?
file_status	tinyint(1)
filepath	char(20)
file_type	char(20)
metadata	text
notes	text

## Projects

project_id	@* int(11)
Name	text?
notes	text

## Pointings

pointing_id	@* int(11)
target_id	int(11)
bf_config_id	int(11)
tobs	float?
utc_start	float
sb_id	text
metadata	text
notes	text

## Processing\_Pivot

processing_pivot_id	@* int(11)
dp_id	int(11)
processing_id	int(11)

## Beams

beam_id	@* int(11)
pointing_id	int(11)
on_target	tinyint(1)
ra	float(10)?
dec	float(10)?
coherent	tinyint(1)?

## Processings

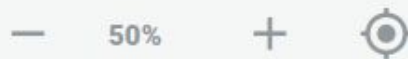
processing_id	@* int(11)
pipeline_id	int(11)
hardware_id	int(11)?
submit_time	float(10)
start_time	float(10)?
end_time	float(10)?
process_status	tinyint(1)?
metadata	text?
notes	text

## Pipelines

pipeline_id	@* int(11)
hash	text
name	text?
notes	text

## Hardwares

hardware_id	@* int(11)
name	text?
metadata	text?
notes	text



**DEMO**