SQL for Data Science

Bharat Bhushan Verma

What is SQL?

- SQL: Structured Query Language is used to talk to
- RDBMS: Relational Database Management Systems characterized by
- ACID: Atomicity Consistency Isolation Durability for
- CRUD: Create Read Update and Delete records or
- WORM: Write Once Ready Many Operations

- SQL is used to communicate with Databases
- It is a non-procedural/declarative language

Different SQL Roles

- Data Administrator
- Data Architect
- ETL Developer
- Backend Developer
- Systems Engineer
- System Admin
- QA Engineer
- **Data Analyst**
- **Data Scientist**

How Data Analyst / Scientists use SQL

- End User of Database
- Use SQL to query and retrieve data
- Occasionally, Create Tables or Add columns in Tables
- Combine Multiple Sources together
- Write Complex queries to create flat tables for analysis
- Note: Sometimes you may write different dialect of SQL depending upon the DBMS.

Dialects of SQL

Popular DBMS

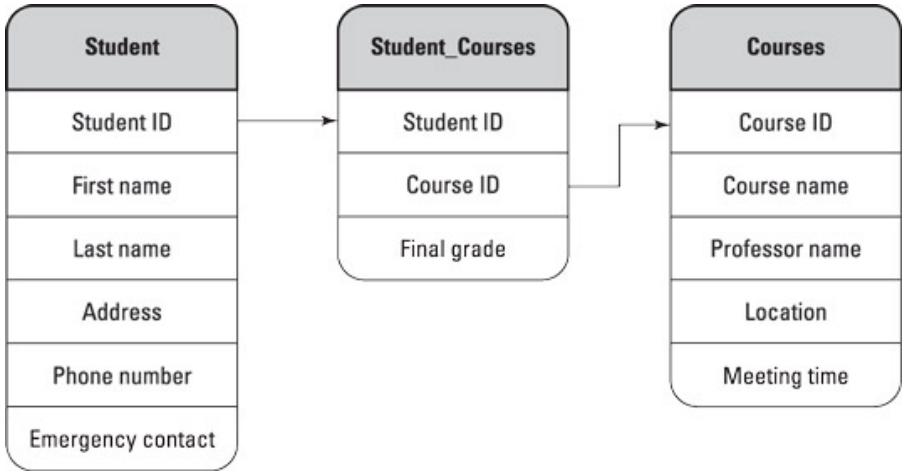
- MS SQL
- IBM DB2 Oracle
- Sybase ASE
- Postgres SQL
- My SQL
- Apache Open Office Base
- SQL Lite
- RISE

Databases as a Service

- https://www.freesqldatabase.com/
- https://aws.amazon.com/
- https://azure.microsoft.com/en-gb/
- https://www.microsoft.com/en-us/sql-server/sql-server-editions-express
- https://www.ibm.com/cloud/databases

What is a Database?

- Database is a collection of tables interlinked with each other on some common field/variable.
- It is like a container (DB) having a set of related files (Tables) which is a structured list of data elements.
- A table consists of columns (field / variable) and rows (record).
-



Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

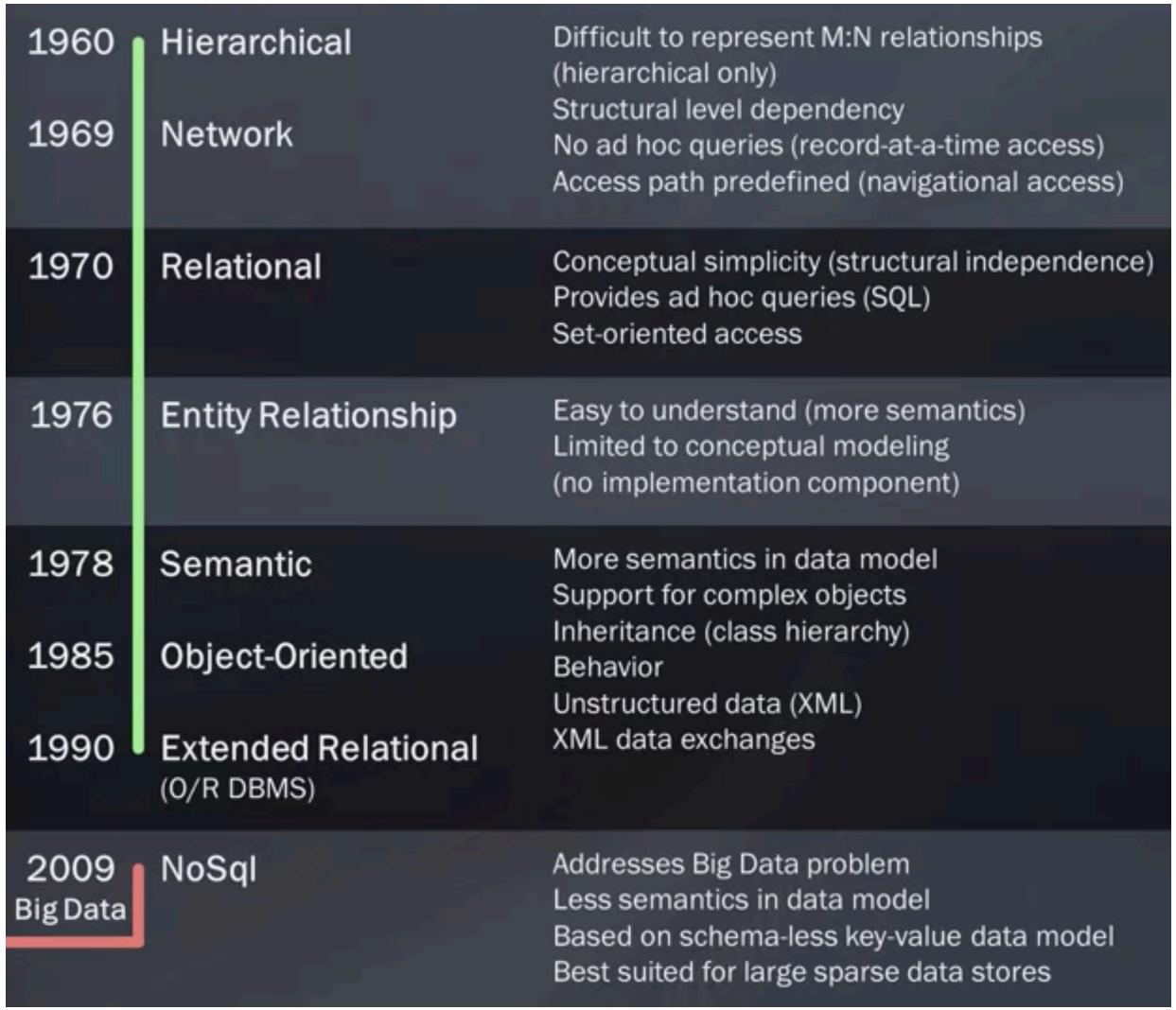
Activities Table

ID*	Activity1	Cost1	Activity2	Cost2
084	Tennis	$36	Swimming	$17
100	Squash	$40	Swimming	$17
182	Tennis	$36		
219	Swimming	$15	Golf	$47

What is Data Modeling

- Organizes and Structures information into related tables
- Represents relationships among tables (Business processes).
- Abstractly represent the real world.
- Types of Data models
  - Data Models for prediction built by data scientists.
  - Data Models for recording transactions in data tables in a DB.

Data Models


1960
Hierarchical
1969
Network
Difficult to represent M:N relationships
(hierarchical only)
Structural level dependency
No ad hoc queries (record-at-a-time access)
Access path predefined (navigational access)
1970
Relational
Conceptual simplicity (structural independence)
Provides ad hoc queries (SQL)
Set-oriented access
1976
Entity Relationship
Easy to understand (more semantics)
Limited to conceptual modeling
(no implementation component)
1978
Semantic
1985
Object-Oriented
1990
Extended Relational
(O/R DBMS)
More semantics in data model
Support for complex objects
Inheritance (class hierarchy)
Behavior
Unstructured data (XML)
XML data exchanges
2009
Big Data
NoSql
Addresses Big Data problem
Less semantics in data model
Based on schema-less key-value data model
Best suited for large sparse data stores

NoSQL is Not Only SQL

Relational data model

Highly-structured table organization with rigidly-defined data formats and record structure.

Document data model

Collection of complex documents with arbitrary, nested data formats and varying "record" format.

- A mechanism for storage and retrieval of unstructured data modeled by means other than tabular relations in relational databases.

SQL Vs NoSQL

Type	SQL	No SQL
Data Storage	Stored in relational model with rows and columns	Refers to family of DBs: Document, Graph, Key-Value, Columnar
Schema	Fixed Schema. Can be altered for whole DB.	Dynamic Schema. Many missing values Can be altered easily.
Scalability	Vertical Scaling. More data means bigger servers. Across servers is difficult.	Horizontal Scaling. More data means multiple cheap commodity servers.
ACID/BASE	ACID Compliant	Sacrifice ACID for performance and scalability.

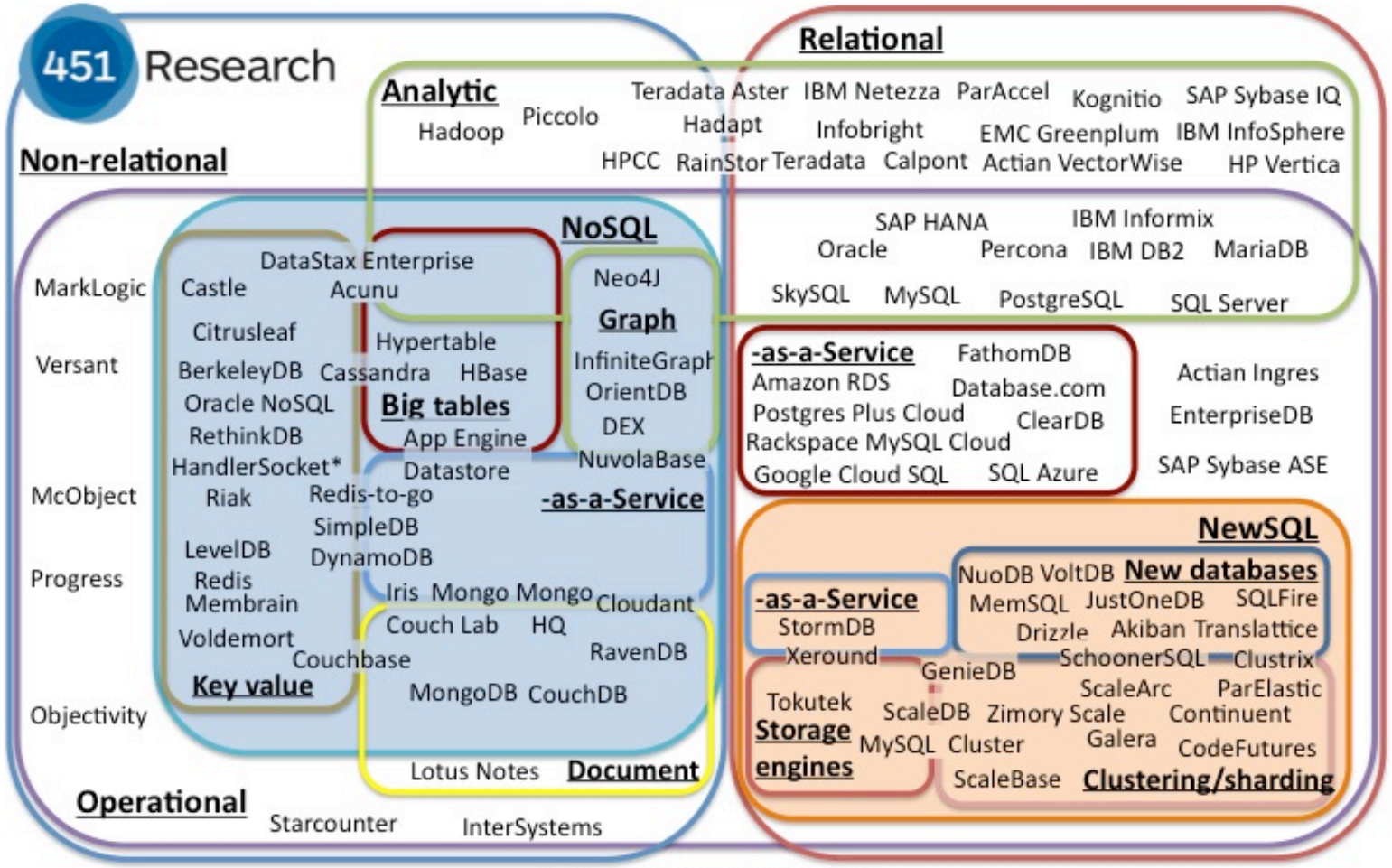
Faces of NoSQL

- Document Databases: Instead of storing data in rows and columns in a table, data is stored in documents, and these documents are grouped together in collections. Each document can have a completely different structure. Document databases include the aforementioned CouchDB and MongoDB.
- Key-Value Stores: Data is stored in an associative array of key-value pairs. The key is an attribute name, which is linked to a value. Well-known key value stores include Redis, Voldemort (developed by LinkedIn) and Dynamo (developed by Amazon).

Faces of NoSQL

- Graph Databases: Used for data whose relations are represented well in a graph. Data is stored in graph structures with nodes (entities), properties (information about the entities) and lines (connections between the entities). Examples of this type of database include Neo4J and InfiniteGraph.
- Columnar Databases: Instead of ‘tables’, in columnar databases you have column families, which are containers for rows. Unlike RDBMS, you don’t need to know all of the columns up front, each row doesn’t have to have the same number of columns. Columnar databases are best suited to analysing huge datasets- big names include Cassandra and HBase.

Evolving Databases landscape



Real-time Intelligent Secure Explainable Systems

- RISELab: a proactive step to move beyond Big Data analytics into a more immersive world. The RISE agenda begins by recognizing that there are big changes afoot:
  - *Sensors are everywhere.* We carry them in our pockets, we embed them in our homes, we pass them on the street. Our world will be quantified, in fine detail, in real time.
  - *AI is for real.* Big data and cheap compute finally made some of the big ideas of AI a practical reality. There's a ton more to be done, but learning and prediction are now practical tools in the computing toolbox.
  - *The world is programmable.* Our vehicles, houses, workplaces and medical devices are increasingly networked and programmable. The effects of computation are extending to include our homes, cities, airspace, and bloodstreams.
- In short, the loop between data generation, computation, and actuation is closing. And this is no longer a niche scenario: it's going to be a standard mode of technology going forward.
- https://rise.cs.berkeley.edu/

Relational and Transactional Models

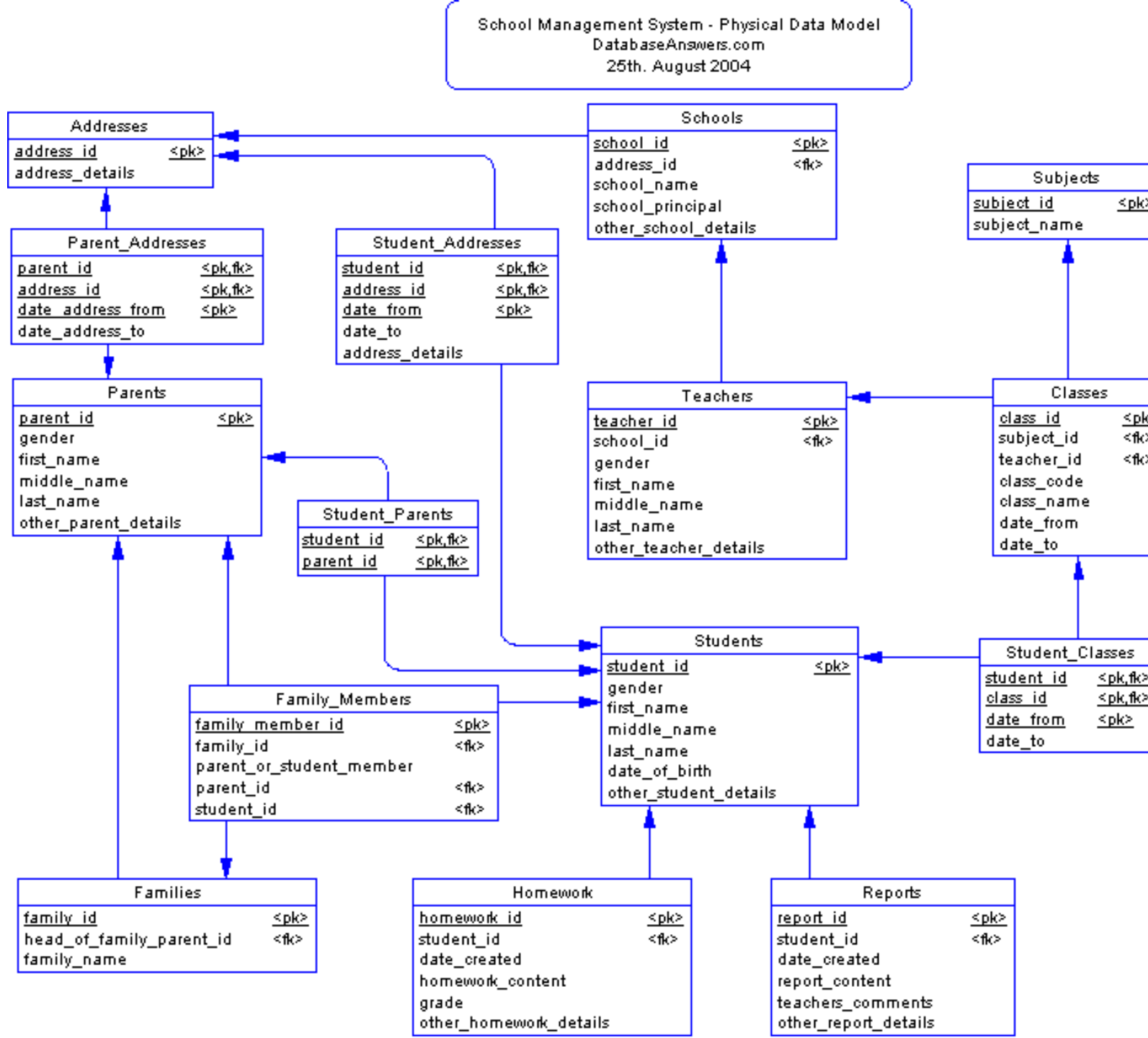
- Transactional models are used for operational database for recording as events happen in the real world. (OLTP)
- Relational models are used for querying, manipulating, archiving and optimizing data in easy and intuitive way (OLAP)

Relational Models

- Entity: Person, Place, Objects, Events etc.
- Attribute: Characteristic of an entity
- Relationship: describes associations among entities
  - One-to-One (Students to Student Contact)
  - One-to-Many (Student to Personal Electronic devices held)
  - Many-to-One (Students to University)
  - Many-to-Many (Students to Courses)
  - Self Joins: (Students as Instructors and Vice versa)

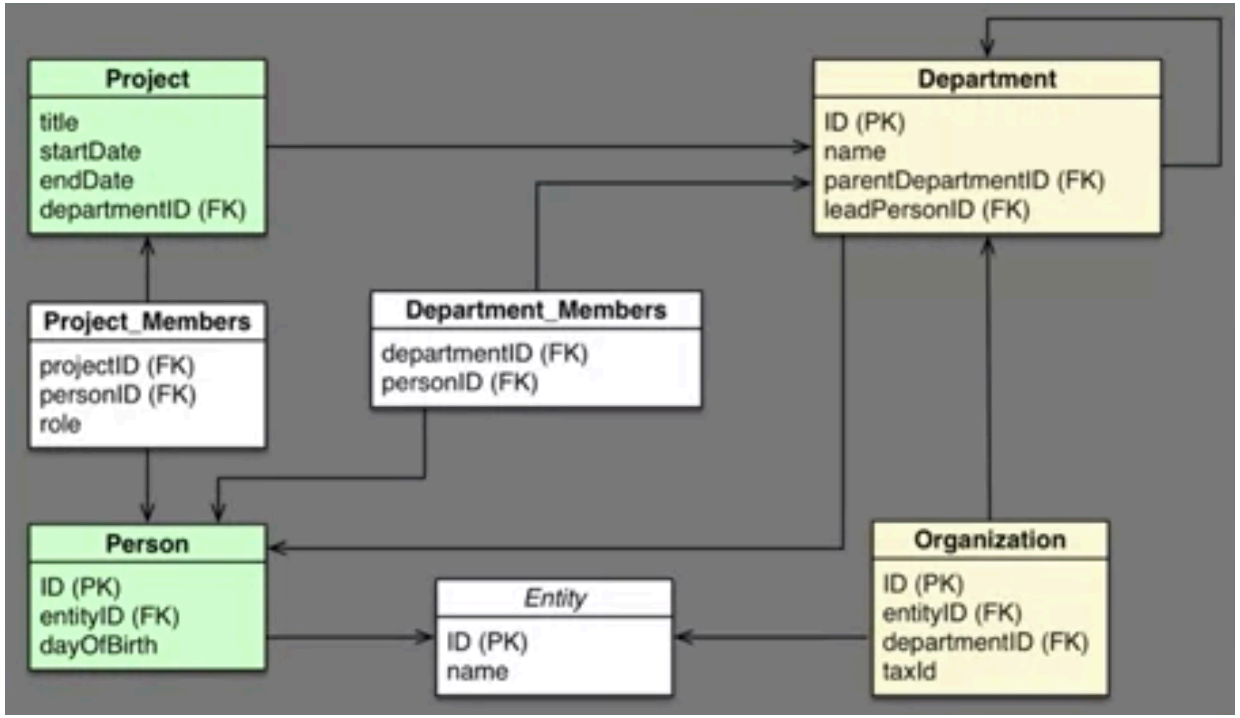
ER Diagram

Show entity relationships
Business Processes
Represented Visually
Show links (Primary Key)



Primary and Foreign Key

- Primary Key: A column or set of columns who values uniquely identify a row in a table.
- Foreign Key: One or more columns that can be used together to identify a single row in another table.



Chen Notations

Chen Notation

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

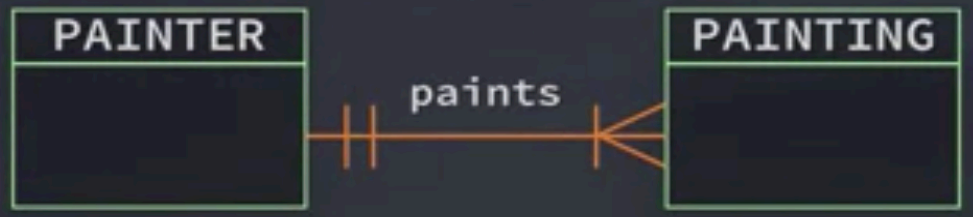
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

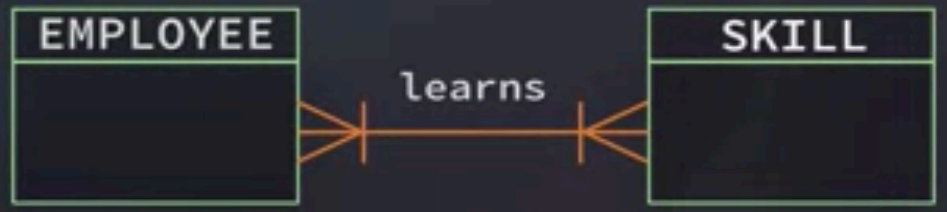
Crows Foot Notations

Crow's Foot Notation

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

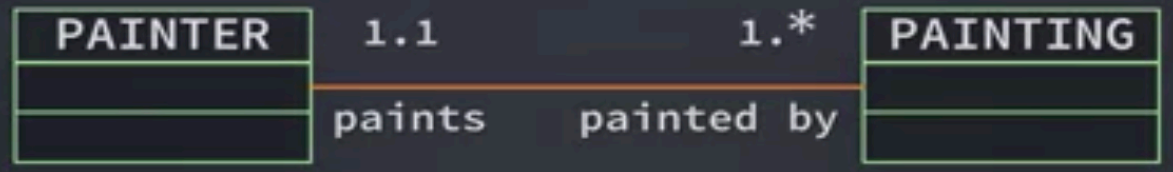


A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

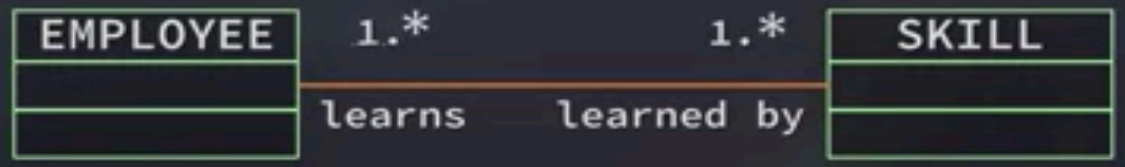
UML Notations

UML Class Diagram Notation

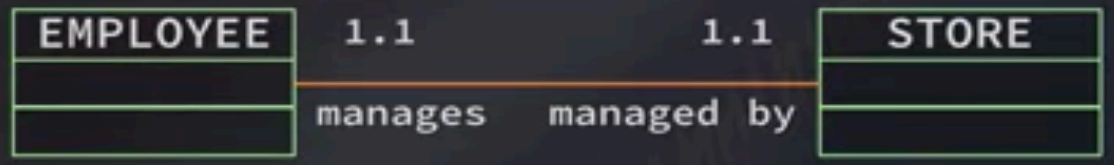
A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.



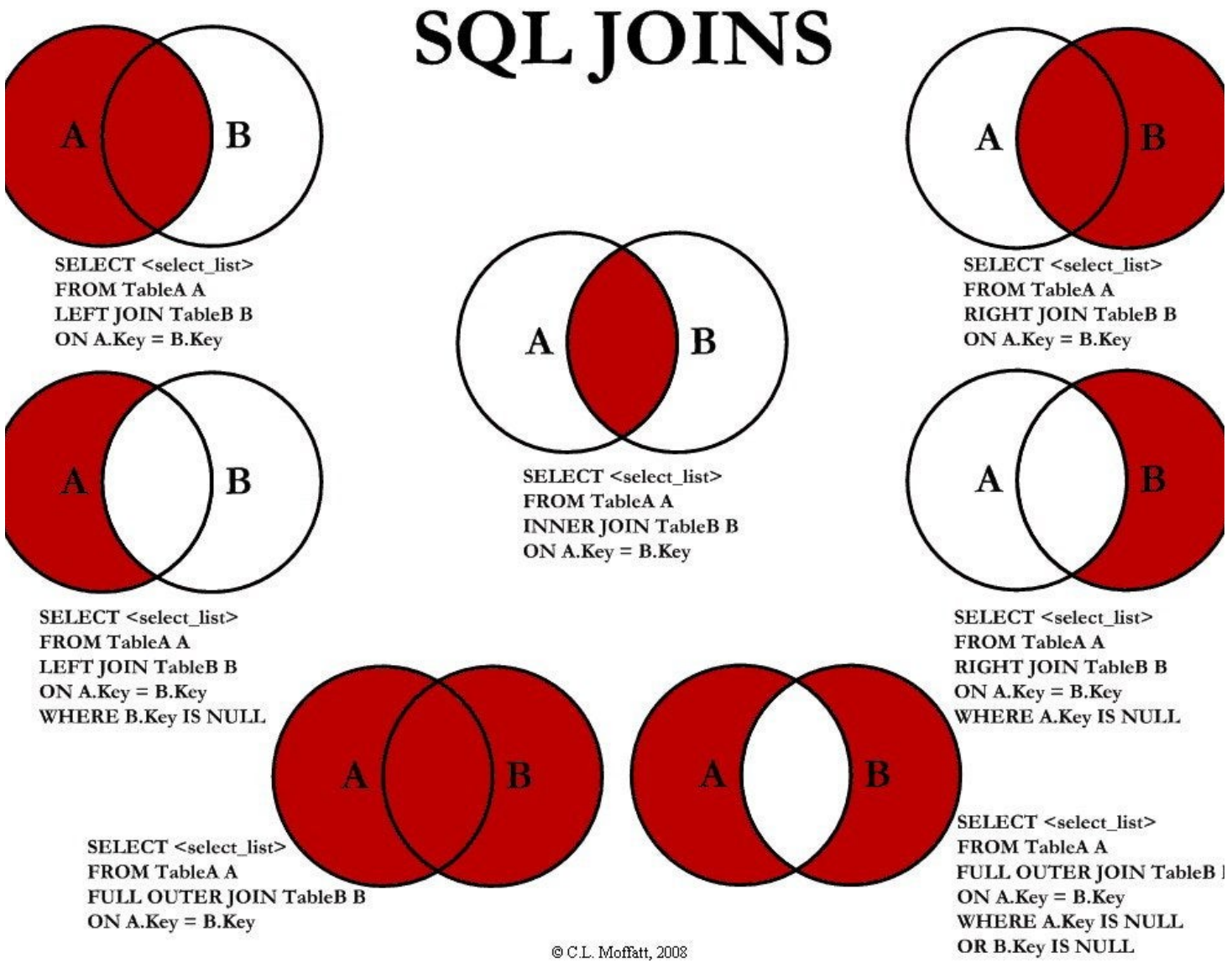
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.



A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.



SQL Joins


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008

Normalization of Databases

- RDBMS are designed to minimize duplications to make database efficient and avoid data conflicts. It is achieved by dividing data into sub tables and create pointers to data through a process called normalization.

Course	Classroom	Professor	Contact
MBA6693	SH351	Bharat	7307
ADM2624	T104	Martin	7999
ADM2623	T125	Bharat	7307

Solution to Avoid Duplicity

Course	Location	Prof ID
MBA6693	SH351	1
ADM2624	T104	2
ADM2623	T125	1

Prof ID	Professor	Contact
1	Bharat	7307
2	Martin	7999

First Normal Form (Atomic Data)

Professor	Courses
Bharat	MBA6693, ADM2623
Martin	ADM2624

Professor	Courses
Bharat	MBA6693
Bharat	ADM2623
Martin	ADM2624

Second Normal Form

- Table should be in 1 NF
- Primary should compose of exactly 1 column

Student	Course
Alex	ADM3628
Prajain	ADM3628
Alex	MBA6693
Aditya	MBA6607

Student	Enrollment
Alex	1
Prajain	2
Aditya	3

Course	Student ID
ADM3628	1
ADM3628	2
MBA6693	1
MBA6607	3

Third Normal Form (~~Functional Dependency~~)

Course	Location	Professor	Department
MBA6693	SH351	Bharat	Business Analytics
ADM2623	T124	Martin	Statistics

Course	Location	Professor	Department
MBA6693	SH351	Donglei	Mathematics
ADM2623	T124	Martin	Statistics

Create Database

- Go to www.freesqldatabase.com
- Register an account
- Create a New database
- You will receive notification when your database is ready
  - Database name, user name, password, port and host address etc.
- Access the database using http://www.phpmyadmin.co

Create Table Employees

```
CREATE TABLE EMPLOYEES (
        EMP_ID CHAR(9) NOT NULL,
        F_NAME VARCHAR(15) NOT NULL,
        L_NAME VARCHAR(15) NOT NULL,
        SSN CHAR(9),
        B_DATE DATE,
        SEX CHAR,
        ADDRESS VARCHAR(30),
        JOB_ID CHAR(9),
        SALARY DECIMAL(10,2),
        MANAGER_ID CHAR(9),
        DEP_ID CHAR(9) NOT NULL,
        PRIMARY KEY (EMP_ID));
```

Create Table Job History

```
CREATE TABLE JOB_HISTORY (
        EMP_ID CHAR(9) NOT NULL,
        START_DATE DATE,
        JOB_ID CHAR(9) NOT NULL,
        DEP_ID CHAR(9),
        PRIMARY KEY (EMP_ID,JOB_ID));
```

Create Table Jobs

```
CREATE TABLE JOBS (
        JOB_ID CHAR(9) NOT NULL,
        JOB_TITLE VARCHAR(15) ,
        MIN_SALARY DECIMAL(10,2),
        MAX_SALARY DECIMAL(10,2),
        PRIMARY KEY (JOB_ID));
```

Create Table Departments

```
CREATE TABLE DEPARTMENTS (
        DEP_ID CHAR(9) NOT NULL,
        DEP_NAME VARCHAR(15) ,
        MANAGER_ID CHAR(9),
        LOC_ID CHAR(9),
        PRIMARY KEY (DEP_ID));
```

Create Table Locations

```
CREATE TABLE LOCATIONS (
        LOC_ID CHAR(9) NOT NULL,
        DEP_ID CHAR(9) NOT NULL,
        PRIMARY KEY (LOC_ID,DEP_ID));
```

INSERT Data

```
INSERT INTO employees
VALUES (‘E001’,
        ’Bharat’,
        ‘Bhushan’,
        ‘123456789’,
        ‘2011-12-12’,
        ‘Male’,
        ‘321 Tilley Hall’,
        ‘J321’,
        ‘100,000.50’,
        ‘M001’,
        ‘D001’);
```

Update and Delete Command

- UPDATE *EMPLOYEES*
  SET *EID = value1, F_Name = value2, ...*
  WHERE *Gender = “Male”*;

- DELETE FROM EMPLOYEES WHERE F_Name=‘XYZ';
-