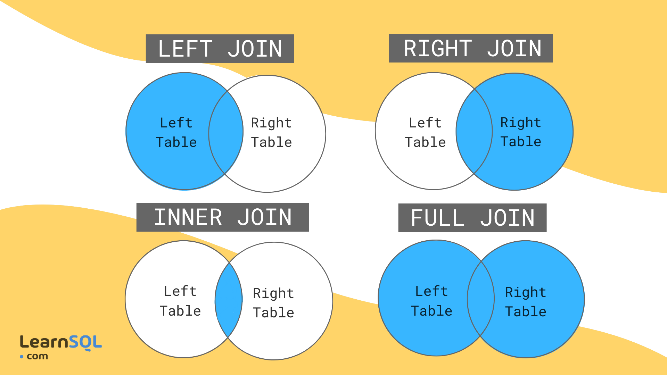
* **Foreign Keys & Index have ()**
* **Joins have a period .**
* **LEFT JOIN 2ndtable ON 1stTable.MatchingColumn=2ndTable.MatchingColumn;**
* **CHANGE COLUMN Year ReleaseYear SMALLINT;**
* **Creating table - FOREIGN KEY (RatingCode) REFERENCES Rating (RatingCode));**
* **Altering table - ALTER TABLE Movie** 
  + - **ADD CONSTRAINT Year**
    - **FOREIGN KEY (Year) REFERENCES YearStats(Year);**

**joins**

* ***INNER JOIN* selects only matching left and right table rows.**
* ***FULL JOIN* selects all left and right table rows, regardless of match.**
* ***LEFT JOIN* selects all left table rows, but only matching right table rows.**
* ***RIGHT JOIN* selects all right table rows, but only matching left table rows.**
* **An *outer join* is any join that selects unmatched rows, including left, right, and full joins.**

**8 Practice Labs:**

A diagram of a number of symbols

Description automatically generated

Table 1.1.2: Similar data structure terms.

|  |  |  |
| --- | --- | --- |
| Databases | Mathematics | Files |
| Table | Relation | File |
| Column | Attribute | Field |
| Row | Tuple | Record |
| Data type | Domain | Data type |
| ALTER TABLE clause | Description | Syntax |
| ADD | Adds a column | ALTER TABLE TableName  ADD ColumnName DataType; |
| CHANGE | Modifies a column | ALTER TABLE TableName  CHANGE CurrentColumnName NewColumnName NewDataType; |
| DROP | Deletes a column | ALTER TABLE TableName  DROP ColumnName; |

SELECT manager\_id, MAX(salary)

FROM employees

GROUP BY manager\_id;

-- Your SQL statements go here

SELECT Movie.Title, Movie.year, Rating.Description

FROM Movie

LEFT JOIN Rating

ON Movie.RatingCode = Rating.Code;

**-- Your SQL goes here**

CREATE TABLE Movie (

Title VARCHAR(30),

RatingCode VARCHAR(5),

FOREIGN KEY (RatingCode) REFERENCES Rating (RatingCode)

);

**-- Your SQL goes here**

ALTER TABLE Movie

ADD CONSTRAINT Year

FOREIGN KEY (Year) REFERENCES YearStats(Year);

**-- Your SQL goes here**

SELECT Movie.Title, YearStats.TotalGross

FROM Movie

LEFT JOIN YearStats ON Movie.Year = YearStats.Year;

**-- Your SQL goes here**

CREATE TABLE Member (

ID INT UNSIGNED,

FirstName VARCHAR(100),

MiddleName CHAR(1),

LastName VARCHAR(100),

DateOfBirth DATE,

AnnualPledge DECIMAL(8,2) UNSIGNED

);

**-- Your SQL goes here**

ALTER TABLE MOVIE

ADD Score DECIMAL(3,1)

;

**-- Your SQL goes here**

CREATE VIEW MyMovies AS

SELECT Title, Genre, Year

From Movie;

**-- Your SQL goes here**

DROP VIEW MovieView;

**-- Your SQL goes here**

ALTER TABLE Movie

ADD PRIMARY KEY (ID);

**-- Your SQL goes here**

CREATE INDEX idx\_year ON Movie(Year);

**-- Your SQL goes here**

INSERT INTO Movie (Title, Genre, RatingCode, Year)

VALUES ('Pride and Prejudice', 'Romance', 'G', 2005);

**-- Your SQL goes here**

DELETE FROM Movie

WHERE ID=3;

**-- Your SQL goes here**

UPDATE Movie

SET Year=2022

WHERE Year=2020;

**-- Your SQL goes here**

SELECT \* FROM Movie;

**-- Your SQL goes here**

SELECT Title, Genre

FROM Movie

WHERE Year=2020;

**-- Your SQL goes here**

SELECT Title

FROM Movie

ORDER BY Title ASC;

**- Your SQL goes here**

SELECT RatingCode, COUNT(\*) AS RatingCodeCount

FROM Movie

GROUP BY RatingCode

ORDER BY RatingCode ASC;

**-- Your SQL goes here**

SELECT COUNT(\*) MovieCount

FROM Movie

WHERE Year = 2019;

**2 Practice Labs:**

-- Your SQL statements go here

SELECT year, COUNT(\*)

FROM Movie

GROUP BY year;

-- Your SQL statements go here

SELECT RegisteredName, Height

FROM Horse

WHERE Height > (SELECT AVG(Height) FROM Horse)

ORDER BY Height ASC;

**7 pRACTICE lABS:**

-- Your SQL statements go here

ALTER TABLE Movie ADD Producer Varchar(50);

ALTER TABLE Movie DROP COLUMN Genre;

ALTER TABLE Movie CHANGE COLUMN Year ReleaseYear SMALLINT;

INSERT INTO Horse (RegisteredName, Breed, Height, BirthDate)

VALUES

('Babe', 'Quarter Horse', 15.3, '2015-02-10'),

('Independence', 'Holsteiner', 16.0, '2017-03-13'),

('Ellie', 'Saddlebred', 15.0, '2016-12-22'),

(NULL, 'Egyptian Arab', 14.9, '2019-10-12')

;

UPDATE Horse

SET Height=15.6

WHERE ID=2;

UPDATE Horse

SET RegisteredName='Lady Luck', BirthDate='2015-05-01'

WHERE ID=4;

UPDATE Horse

SET Breed=Null

WHERE BirthDate>='2016-12-22';

DELETE FROM Horse

WHERE ID=5;

DELETE FROM Horse

WHERE Breed IN('Holsteiner', 'Paint');

DELETE FROM Horse

WHERE BirthDate<('2013-03-13');

SELECT RegisteredName, Height, BirthDate

FROM Horse

WHERE (Height BETWEEN 15.0 AND 16.0)

OR (BirthDate >= '2020-01-01');

CREATE TABLE Movie(

ID SMALLINT UNSIGNED,

Title VARCHAR(50),

Rating CHAR(4),

ReleaseDate DATE,

Budget DECIMAL(8,2));

CREATE TABLE Student (

ID SMALLINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(20) NOT NULL,

LastName VARCHAR(30) NOT NULL,

Street VARCHAR(50) NOT NULL,

City VARCHAR(20) NOT NULL,

State CHAR(2) NOT NULL DEFAULT 'TX',

Zip MEDIUMINT UNSIGNED NOT NULL,

Phone CHAR(10) NOT NULL,

Email VARCHAR(30) UNIQUE

);

CREATE TABLE Horse (

ID SMALLINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

RegisteredName VARCHAR(15) NOT NULL,

Breed VARCHAR(20) CHECK (Breed IN ('Egyptian Arab', 'Holsteiner', 'Quarter Horse', 'Paint', 'Saddlebred')),

Height DECIMAL(3,1) CHECK (Height >= 10.0 AND Height <= 20.0),

BirthDate DATE CHECK (BirthDate >= '2015-01-01')

);