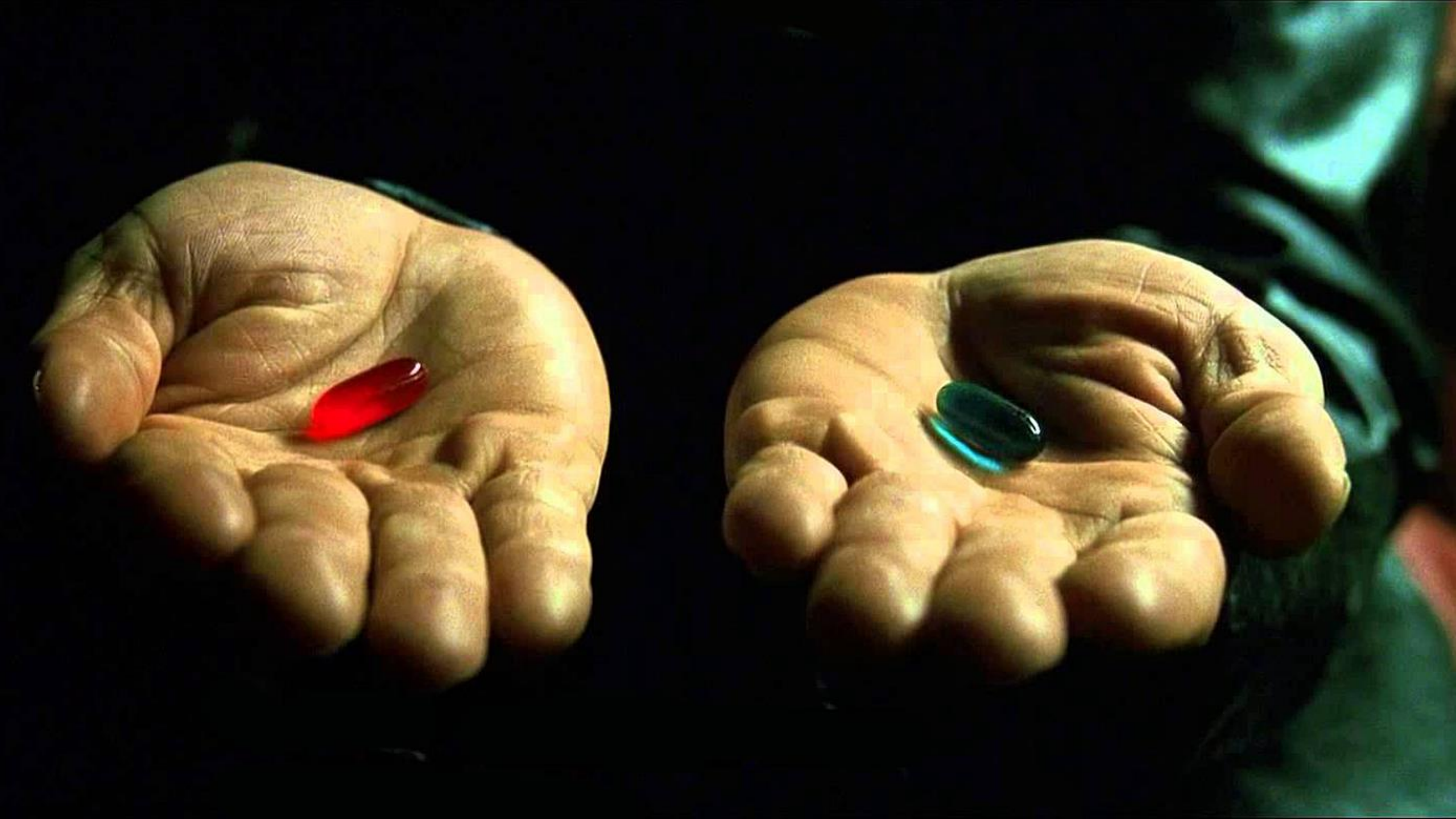

Advanced SQL × SELECT

5 SELECT DISTINCT Columns
1 FROM Tables
2 [WHERE θ]
3 [GROUP BY Columns]
4 [HAVING θ']
6 [ORDER BY Columns [ASC | DESC]]
7 [LIMIT # [OFFSET #]]

Advanced SQL × SELECT

2 SELECT Columns
1 FROM Table

Advanced SQL × SELECT

3 SELECT Columns
1 FROM Table
2 [WHERE θ]

Advanced SQL × DISTINCT

3 SELECT DISTINCT Columns
1 FROM Table
2 [WHERE θ]

To eliminate duplicate tuples, considering all columns.

Advanced SQL × DISTINCT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

On what years at least one movie has been released?

Advanced SQL × DISTINCT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

On what years at least one movie has been released?

```
SELECT ReleaseDate
FROM Movie
```

ReleaseDate
1968
1968
1963
1968

Advanced SQL × DISTINCT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

In what years at least one movie has been released?

```
SELECT DISTINCT ReleaseDate
FROM Movie
```

ReleaseDate
1968
1963

Advanced SQL × DISTINCT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

In what years at least one movie has been released?

```
SELECT DISTINCT ReleaseDate, Language
FROM Movie
```

ReleaseDate	Language
1968	*English*
1963	*English*
1968	*EN*

Advanced SQL × DISTINCT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

What languages have been used in movies?

```
SELECT DISTINCT Language
FROM Movie
```

Language
English
EN

Advanced SQL × Math Operation

3 SELECT Column {+, -, *, /, %, ...} Column | Constant
1 FROM Tables
2 [WHERE θ]

To apply a function on each value of a column.

Advanced SQL × Math Operation

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

List movies and their length in HH:MM?

```
SELECT Title,
       RunningTime / 60  AS Hour,
       RunningTime % 60 AS Minutes
FROM Movie
```

Title	Hour	Minutes
2001: A Space Odyssey	*2*	*22*
Rosemary's Baby	NULL	NULL
The Birds	*1*	*59*
Planet of the Apes	*1*	*52*

Advanced SQL × Built-in Function

3 SELECT FUNCTION(Column), ...
1 FROM Tables
2 [WHERE θ]

To apply a function on each value of a column.

Advanced SQL × Built-in Function 30

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

List movies and their length in HH:MM?

```
SELECT Title, TIME(RunningTime * 60, 'unixepoch') AS Length
FROM Movie
```

Title	Length
2001: A Space Odyssey	*2:22:00*
Rosemary's Baby	NULL
The Birds	*01:59:00*
Planet of the Apes	*01:52:00*

Advanced SQL × Built-in Function 31

Refer to DBMS's manual for more built-in functions.

SQLite → https://www.sqlite.org/lang_corefunc.html

Advanced SQL × AGG Function

3 SELECT COUNT | SUM | MAX | MIN | AVG(Column)
1 FROM Tables
2 [WHERE θ]

To apply AGGregation functions on non-NULL values of one column and return a single value.

Advanced SQL × COUNT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many movies have been made in 1968?

```
SELECT COUNT(Title) AS MovieCount
FROM Movie
WHERE ReleaseDate = 1968
```

MovieCount
3

Advanced SQL × COUNT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many movies have been made in 1968?

```
SELECT COUNT(Language) AS MovieCount
FROM Movie
WHERE ReleaseDate = 1968
```

MovieCount
3

Advanced SQL × COUNT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many movies have been made in 1968?

```
SELECT COUNT(RunningTime) AS MovieCount
FROM Movie
WHERE ReleaseDate = 1968
```

MovieCount
2

Advanced SQL × COUNT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many movies have been made in 1968?

```
SELECT COUNT(*) AS MovieCount
FROM Movie
WHERE ReleaseDate = 1968
```

MovieCount
3

Advanced SQL × AVG

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

On average, what's the length of movies have been made in 1968?

```
SELECT AVG(RunningTime) AS AveMovieLength
FROM Movie
WHERE ReleaseDate = 1968
```

AveMovieLength
183

Advanced SQL × MIN

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

What's the shortest length of movies have been made in 1968?

```
SELECT MIN(RunningTime) AS MinMovieLength
FROM Movie
WHERE ReleaseDate = 1968
```

MinMovieLength
112

Advanced SQL × MAX

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

What's the longest length of movies have been made in 1968?

```
SELECT MAX(RunningTime) AS MaxMovieLength
FROM Movie
WHERE ReleaseDate = 1968
```

MaxMovieLength
142

Advanced SQL × SUM

Advanced SQL × AGG Function

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

What's the shortest, longest, and average length of movies have been made in 1968?

```
SELECT MIN(RunningTime)     AS Min,
       MAX(RunningTime)     AS Max,
       AVG(RunningTime)     AS Avg,
       SUM(RunningTime)     AS Sum,
       COUNT(RunningTime)   AS Count,
       COUNT(*)
FROM Movie
WHERE ReleaseDate = 1968
```

Min	Max	Avg	Sum	Count	Count(*)
112	*142*	*183*	*254*	*2*	*3*

Advanced SQL × AGG × DISTINCT 42

3 SELECT COUNT | SUM | MAX | MIN | AVG(DISTINCT Column)
1 FROM Tables
2 [WHERE θ]

To apply AGG functions on non-NULL values of one column, after removing duplicates, and return a single value.

Advanced SQL × AGG × DISTINCT

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many different languages have been used in movies?

```
SELECT COUNT(DISTINCT Language) AS Count
FROM Movie
```

Count
2

Advanced SQL × AGG × Math

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

What's the longest movie in hour which have been made in 1968?

```
SELECT MAX(RunningTime / 60) AS Max,
FROM Movie
WHERE ReleaseDate = 1968
```

Max
2

Advanced SQL × AGG × Built-in

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many years old is the oldest movie?

```
SELECT MAX(STRFTIME('%Y', 'now') – ReleaseDate) AS Result
FROM Movie
WHERE ReleaseDate = 1968
```

Result
56

Advanced SQL × WHERE

3 SELECT Columns
1 FROM Tables
2 [WHERE θ]

Advanced SQL × WHERE × Math

Advanced SQL × WHERE × Built-in

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

List all movies which are older than 55 years?

```
SELECT * FROM Movie
WHERE STRFTIME('%Y', 'now') – ReleaseDate > 55
```

Advanced SQL × WHERE × ~~AGG~~

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

What are the longest movies which have been made in 1968?

```
SELECT *
FROM Movie
WHERE ReleaseDate = 1968 AND
      ~~RunningTime = MAX(RunningTime)~~
```

Advanced SQL × WHERE × ~~AGG~~

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

What is the oldest movie?

```
SELECT *
FROM Movie
WHERE ~~ReleaseDate = MIN(ReleaseDate)~~
```

Advanced SQL × WHERE × NULL 50

NULL represents two facts about information

I) No Value, e.g., non-American directors do not have SSN
II) Missing | Unknow Value, e.g., PlaceOfBirth for a director might be missed, or unknown at the time of data entrance,

Advanced SQL × WHERE × NULL

Comparing with NULL value result in NULL, not FALSE, not TRUE!

```
SELECT NULL=1  AS Result;
SELECT NULL<>1 AS Result;
SELECT NULL>1  AS Result;
SELECT NULL>=1 AS Result;
...
```

Result
NULL

Advanced SQL × WHERE × IS NULL 52

To compare with NULL value explicitly, IS NULL | IS NOT NULL:

```
SELECT NULL IS NULL     AS Result1,
       NULL IS NOT NULL AS Result2;
```

Result1	Result2
TRUE	FALSE

Advanced SQL × WHERE × IS NULL

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many movies whose running time is missing?

```
SELECT COUNT(*) AS IncompleteMovieCount
FROM Movie
WHERE RunningTime IS NULL
```

IncompleteMovieCount
1

Advanced SQL × WHERE × LIKE 54

To compare with CHAR-based values symbol, SQL has LIKE | NOT LIKE operator and pattern matching symbols:

I) %, Represents 0 or more of any CHAR, called wildcard

II) _ , Represents any single character

Advanced SQL × WHERE × LIKE

e.g.,

LIKE 'H%' : matches values start with 'H' or 'h', the rest can be anything.

LIKE 'H_ _ _' : matches values with exactly four CHARs starting with 'H' or 'h'.

LIKE '%e' : matches values ends with 'e' or 'E'.

LIKE '%birds%': matches values containing 'birds', or 'BiRds'.

NOT LIKE 'H%': matches values whose first character is not 'H' or 'h'.

LIKE operator is not case sensitive for ASCII chars

Advanced SQL × WHERE × LIKE

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

Find the movie 'space odyssey'?

```
SELECT *
FROM Movie
WHERE Title LIKE '%space odyssey%'
```

Advanced SQL × WHERE × LIKE

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

Find the movie 'space odyssey'?

```
SELECT *
FROM Movie
WHERE Title = '%space odyssey%'
```

Advanced SQL × FROM

3 SELECT Columns
1 FROM Tables
2 [WHERE θ]

Advanced SQL × FROM × Product(×) 59

3. SELECT Columns
1. FROM *Table1, Table2, …, TableN*
2. WHERE θ

$$\pi_{<Columns>}(\sigma_{\theta}(Table1 \times Table2 \times \ldots \times TableNameN))$$

Advanced SQL × FROM × Product(×) 60

Movie				MovieGenre		Genre	
Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

Find genres of the movie '2001: A Space Odyssey'?

$$\sigma_{\text{Genre.Id=GenreId}}(\sigma_{\text{Movie.Id=MovieId AND Title='2001: A Space Odyssey'}}(\text{Movie} \times \text{MovieGenre})) \times \text{Genre})$$

Advanced SQL × FROM × Product(×) 61

Movie				MovieGenre		Genre	
Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	1	1	1	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	1	3	3	*Adventure*

Find genres of the movie '2001: A Space Odyssey'?

$$\sigma_{\text{Genre.Id=GenreId}}(\sigma_{\text{Movie.Id=MovieId AND Title='2001: A Space Odyssey'}}(\text{Movie}\times\text{MovieGenre}))\times\text{Genre})$$

I) Corollary: $\sigma_{\theta}(\sigma_{\theta'}(\sigma_{\theta''\ldots}(\sigma_{\theta'''}(R)) = \sigma_{\theta \text{ AND } \theta' \text{AND } \theta''\ldots \text{ AND } \theta'''}(R)$

II) Product has commutative property, i.e., $(R1\times R2)\times R3 = R1\times(R2\times R3)=R1\times R2\times R3$

Advanced SQL × FROM × Product(×) 62

Movie				MovieGenre		Genre	
Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

$$\sigma_{\substack{\text{Movie.Id=MovieId AND}\\ \text{Genre.Id=GenreId AND}\\ \text{Title='2001: A Space Odyssey'}}}(\text{Movie}\times\text{MovieGenre}\times\text{Genre})$$

Advanced SQL × FROM × Product(×) 63

Movie				MovieGenre		Genre	
Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

$$\sigma_{\substack{\text{Movie.Id=MovieId AND}\\ \text{Genre.Id=GenreId AND}\\ \text{Title='2001: A Space Odyssey'}}}(\text{Movie}\times\text{MovieGenre}\times\text{Genre})$$

```
SELECT *
FROM Movie, MovieGenre, Genre
WHERE Movie.Id    =MovieId AND
      Genre.Id    =GenreId AND
      Title       ='2001: A Space Odyssey'
```

Relational Algebra × θ-Join

$\bowtie_\theta$, θ-join, is product (×) of relations followed by selection (σ)

$$R1 \bowtie_\theta R2 = \sigma_\theta (R1 \times R2)$$

Advanced SQL × FROM × INNER JOIN 64

3 SELECT Columns
1 FROM Table1, Table2
2 WHERE θ

σ_θ (R1 × R2)

3 SELECT Columns
1 FROM Table1
2 INNER JOIN Table2 ON θ

R1 $\bowtie_\theta$ R2

Advanced SQL × FROM × INNER JOIN

Movie				MovieGenre		Genre	
Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

$\sigma_{\text{Title='2001: A Space Odyssey'}}((\text{Movie} \bowtie_{\text{Movie.Id=MovieId}} \text{MovieGenre}) \bowtie_{\text{Genre.Id=GenreId}} \text{Genre})$

```
SELECT *
FROM Movie
INNER JOIN MovieGenre ON Movie.Id = MovieId
INNER JOIN Genre ON Genre.Id = GenreId
WHERE Title='2001: A Space Odyssey'
```

Advanced SQL × FROM × INNER JOIN 66

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	NULL	47
3	Clint	Eastwood	May 31, 1930	USA	NULL	35

Movie			
Id	Title	Language	RunningTime
1	2001: A Space Odyssey	English	142
2	Rosemary's Baby	English	NULL

What are directors' best movie name?

```
SELECT D.Id, FirstName, LastName, BestMovieId, M.Id, Title
FROM Director AS D, Movie AS M
WHERE M.Id = BestMovieId
```

Advanced SQL × FROM × INNER JOIN

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	NULL	47
3	Clint	Eastwood	May 31, 1930	USA	NULL	35

Movie			
Id	Title	Language	RunningTime
1	2001: A Space Odyssey	English	142
2	Rosemary's Baby	English	NULL

What are directors' best movie name?

```
SELECT D.Id, FirstName, LastName, BestMovieId, M.Id, Title
FROM Director AS D
INNER JOIN Movie AS M ON M.Id = BestMovieId
```

Advanced SQL × FROM × INNER JOIN

Id	FirstName	LastName	BestMovieId	Id	Title
1	*Stanley*	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*

What are directors' best movie name?

```
SELECT D.Id, FirstName, LastName, BestMovieId, M.Id, Title
FROM Director AS D
INNER JOIN Movie AS M ON M.Id = BestMovieId
```

Advanced SQL × FROM × LEFT JOIN

FirstName	LastName	BestMovieId	Id	Title
Stanley	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*
Alfred	*Hitchcock*	NULL	NULL	NULL
Clint	*Eastwood*	NULL	NULL	NULL

What are directors' best movie name if any?

Director $\ltimes_{\text{BestMovieId=Movie.Id}}$ Movie

```
SELECT D.Id, FirstName, LastName, BestMovieId, M.Id, Title
FROM Director AS D
LEFT [OUTER] JOIN Movie AS M ON M.Id = BestMovieId
```

Advanced SQL × FROM × RIGHT JOIN

FirstName	LastName	BestMovieId	Id	Title
Stanley	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*
NULL	NULL	NULL	*2*	*Rosemary's Baby*

List all movies and identify whether each one is the best of its director?

Director $\bowtie_{\text{BestMovieId=Movie.Id}}$ Movie

```
SELECT D.Id, FirstName, LastName, BestMovieId, M.Id, Title
FROM Director AS D
RIGHT [OUTER] JOIN Movie AS M ON M.Id = BestMovieId
```

Advanced SQL × FROM × FULL JOIN

FirstName	LastName	BestMovieId	Id	Title
Stanley	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*
Alfred	*Hitchcock*	NULL	NULL	NULL
Clint	*Eastwood*	NULL	NULL	NULL
NULL	NULL	NULL	*2*	*Rosemary's Baby*

Director $\bowtie_{\text{BestMovieId=Movie.Id}}$ Movie

```
SELECT D.Id, FirstName, LastName, BestMovieId, M.Id, Title
FROM Director AS D
FULL [OUTER] JOIN Movie AS M ON M.Id = BestMovieId
```

Advanced SQL × FROM × RIGHT JOIN

Advanced SQL × FROM × FULL JOIN

NOT currently supported in SQLite!

Think about a workaround & bring it with you next week.

Advanced SQL × SELECT

5 SELECT DISTINCT Columns
1 FROM Tables
2 [WHERE θ]
3 [GROUP BY Columns]
4 [HAVING θ']

Advanced SQL × GROUP BY

To group tuples based on values on columns, usually followed by AGG functions.

Advanced SQL × GROUP BY

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*English*	*1968*	*112*

How many movies have been made in each year?

```
SELECT ReleaseDate, COUNT(*)
FROM Movie
GROUP BY ReleaseDate
```

ReleaseDate	COUNT(*)
1968	*3*
1963	*1*

Advanced SQL × GROUP BY

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many movies have been made in each year per language?

```
SELECT ReleaseDate, Language, COUNT(*)
FROM Movie
GROUP BY ReleaseDate, Language
```

ReleaseDate	Language	COUNT(*)
1968	*English*	*2*
1968	*EN*	*1*
1963	*English*	*1*

Advanced SQL × GROUP BY

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*EN*	*1968*	*112*

How many movies have been made in each year per language?

```
SELECT Title, ReleaseDate, Language, COUNT(*)
FROM Movie
GROUP BY ReleaseDate, Language
```

ReleaseDate	Language	COUNT(*)
1968	*English*	*2*
1968	*EN*	*1*
1963	*English*	*1*

SELECT clause only accepts AGG or columns in GROUP BY list.

Advanced SQL × GROUP BY × HAVING 78

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
3	*The Birds*	*English*	*1963*	*119*
4	*Planet of the Apes*	*English*	*1968*	*112*

In which years more than 2 movies have been released?

```
SELECT ReleaseDate, COUNT(*)
FROM Movie
GROUP BY ReleaseDate
HAVING COUNT(*) > 2
```

ReleaseDate	COUNT(*)
1968	*3*

HAVING only accepts AGG or columns in GROUP BY list.

Advanced SQL × Sorting

5 SELECT DISTINCT Columns
1 FROM Tables
2 [WHERE θ]
3 [GROUP BY Columns]
4 [HAVING θ']
6 [ORDER BY Columns [ASC | DESC]]

Advanced SQL × Sorting

Movie				
Id	Title	Language	ReleaseDate	RunningTime
2	*Rosemary's Baby*	*English*	*1968*	NULL
4	*Planet of the Apes*	*English*	*1968*	*112*
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
3	*The Birds*	*English*	*1963*	*119*

List movies sorted by release date & running time?

```
SELECT *
FROM Movie
ORDER BY ReleaseDate DESC, RunningTime
```

Advanced SQL × Paging

5 SELECT DISTINCT Columns
1 FROM Tables
2 [WHERE θ]
3 [GROUP BY Columns]
4 [HAVING θ']
6 [ORDER BY Columns [ASC | DESC]]
7 [LIMIT # [OFFSET #]]

Advanced SQL × Paging

Movie				
Id	Title	Language	ReleaseDate	RunningTime
1	*2001: A Space Odyssey*	*English*	*1968*	*142*
2	*Rosemary's Baby*	*English*	*1968*	NULL
4	*Planet of the Apes*	*English*	*1968*	*112*
3	*The Birds*	*English*	*1963*	*119*

Top-2 recent movies after skipping the most recent one?

```
SELECT *
FROM Movie
ORDER BY ReleaseDate DESC
LIMIT 2 OFFSET 1
```