*Entity-Relational Model

- Entities : things, process to be modeled
- Relationships : connection between entities
- Attributes : additional info for entities

Symbol	Description	Narrative
	Minimum zero Maximum one (optional)	"May be ... one and only one" "Zero or one"
	Minimum one Maximum one (mandatory)	"Must be ... one and only one"
	Minimum zero Maximum many (optional)	"May be ... one or more" "Zero or more"
	Minimum one Maximum many (mandatory)	"Must be ... one or more than one"

*Data Query Language

5 SELECT [DISTINCT] columns → with ① expressions ② literal ③ function
→ boolean
numeric
date and time
運算式

1 FROM database

2 WHERE

3 GROUP BY

4 HAVING : filter for grouped column.

7 UNION

8 ORDER BY DESC (大到小)/ASC (小到大)

9 LIMIT

```sql
--Calculate a running total by market_date of total purchase quantity for each product, sorted by product id and market_date.
SELECT
    DISTINCT product_id,
    market_date,
    SUM(quantity) OVER (PARTITION BY product_id ORDER BY market_date) AS running_total_quantity
FROM
    customer_purchases
ORDER BY
    product_id, market_date;
```

```sql
-- Divide booths into 5 groups based on the number of times they've been purchased. Identify which group each booth falls into.
SELECT
    booth_number,
    NTILE(5) OVER(ORDER BY COUNT(*) DESC) AS groups
FROM vendor_booth_assignments
GROUP BY booth_number;
```

*Data Model Development

- keys [Primary keys : uniquely identify in same table
  Foreign keys : uniquely identify in another table
  → implement relationships between tables
  - composite keys : 2+ attributes to identify
  - [Natural keys : primary key found in natural world
    Surrogate keys : automatically generate in DB

- identify entity
  - independence
  - identifiable
  - relationship

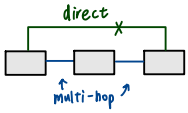
- weak entity : needs to have FK as part of PK

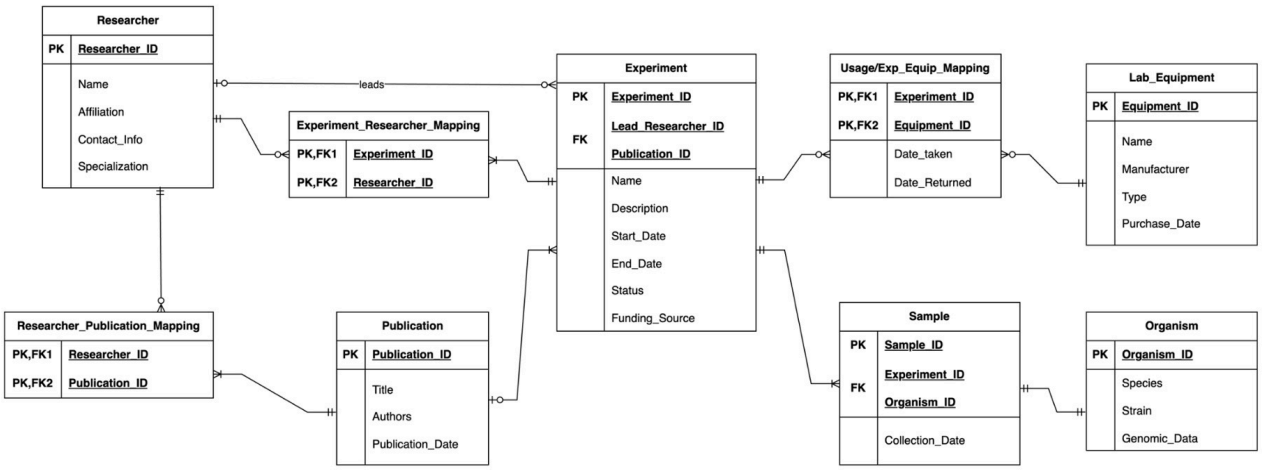
*ER Diagram Development Approach

① Decide entity and attribute
② Draw initial ER (include entity and relationship)
③ Resolve many-to-many relationship
④ Assign PK and FK
⑤ Give Attribute

*Redundant Relationship

- keep "multi-hop" relationship
- remove direct relationship





- WHERE : filter data before group by
  - comparison operators : =, !=, >, <, <>, like, in, between
  - arithmetic operators : +, -, *, /

① Range condition (在A和B之間)
→ column >= A AND column =< B
→ column BETWEEN A AND B
△如果是between文字串
e.g. BETWEEN "FA" AND "FD", which means
字串為fa, fb, fc, fd都會列出

② Member condition (在一個list之中)
→ column IN (A, B, C)
→ column (NOT) IN (SELECT column FROM table WHERE column condition)

③ Partial matching condition
[_ : 只有1個character
% : any numbers of character (0也可以)
→ column LIKE "F%"

Search Expression	Interpretation
F%	Strings beginning with *F*
%t	Strings ending with *t*
%bas%	Strings containing the substring 'bas'
_ _t_	Four-character strings with a *t* in the third position
___-_-____	11-character strings with dashes in the fourth and seventh positions

△Also supports Regular Expression

△Difference between IS and =
① NULL值：用 IS NULL來判斷
② type：IS嚴格check (type一致)；= 會進行type轉換

- CASE : return value based on certain condition

```
CASE WHEN condition1 THEN value
     WHEN condition2 THEN value
     ⋮
     ELSE value
END
```

- Subquery
→ return ① 1 row 1 colum → for comparison
② 1 row multiple column → for IN/NOT IN
③ multiple row multiple column → for FROM table

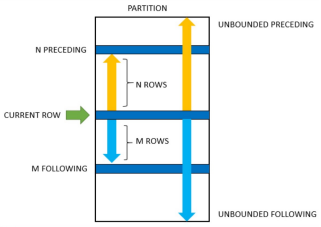
- CTE (Common Table Expressions)
→ define a temporary result, improve readability and maintenance

```
WITH cteName1 AS ( subquery content),
     cteName2 AS ( subquery content)
SELECT
  ⋮
(main query)
```

- Date and Time
  - DATE("YYYY-MM-DD")
  - TIME("HH:MM:SS")
  - TIMESTAMP("YYYY-MM-DD","HH:MM:SS")
  - INTERVAL : add/subtract values e.g. DATE('2020-08-08') + INTERVAL '2' day
  - TIMESTAMP_LTZ America/Los_Angeles TIMESTAMP_NTZ (no time zone) TIMESTAMP_TZ +01:00
  - CURRENT_DATE( ), CURRENT_TIME( ), CURRENT_TIMESTAMP( )
  - date/time extraction : YEAR( ), MONTH( ), DAY( ), HOUR( ), MINUTE( ), SECOND( ), DAYOFWEEK( ), DAYOFYEAR( ), WEEK( ), QUARTER( )
  - formatting and parsing : TO_DATE( ), TO_TIMESTAMP( ), DATE_TRUNC( ) △YEAR(column) = EXTRACT(YEAR column)
  - arithmetic : DATE_DIFF(date1, date2), DATEADD(date, INTERVAL ±value unit)

- WINDOWS FUNCTION
  - common use : running totals, diff between rows, find rank within category

window function OVER(PARTITION column ORDER BY column frame)
根據指定欄分類，特畫面以類別呈現 在類別中以指定欄排序 顯現哪N~M row



- frame
ROWS BETWEEN 3 PRECEDING AND 2 FOLLOWING
RANGE
[ROWS 指定行，不受分類影響 → 可指定前後Nrow
RANGE 指定分類
| UNBOUNDED N | | PRECEDING FOLLOWING | or CURRENT ROW

① Ranking Function : return relative position in partition
- RANK( ) : 排名, e.g. 1, 2, 2, 2, 5, 6, 7...
- DENSE_RANK( ) : 排名but不跳過tie e.g. 1, 2, 2, 2, 3, 4, 5...
- ROW_NUMBER( ) : order by完後顯示的row數 e.g. 1, 2, 3, 4, 5, 6, 7...
- PERCENT_RANK( ) : 第1為1，最後為0，中間是relative percent e.g. 1, 0.8, 0.6, 0.4, 0.2, 0
- NTILE(N) : 將當前partition依序分為N個group e.g. 假設有10 row NTILE(5) → 1, 1, 2, 2, 3, 3, 4, 4, 5, 5

② Value Function : return single value in a relative row
- FIRST_VALUE( ) : return partition內第1個value
- LAST_VALUE( ) : return partition內最後1個value
- LAG(column, 1, NULL) : return 當前row的前1個value △現在為lag延遲，所以取前1
第N個，沒有回傳的值
- LEAD(column, 1, NULL) : return 當前row的下1個value △現在為lead超前，所以取後1
- NTH_VALUE(N) : return 第n row的value

③ Aggregate Function : return value of aggregation but won't be grouped as group by.
SUM( ), AVG( ), MIN( ), MAX( ), COUNT( )

- JOIN (串column)
  - INNER JOIN (default) table ON column = column
    △order of join doesn't matter
  - LEFT JOIN / RIGHT JOIN / FULL JOIN

- UNION (串row)
  - UNION
  - UNION ALL: keep duplicate
  - INTERECT : 取交集(2個table都有的)
  - EXCEPT