SERIOUS SQL LIVE WEEK 6: 5TH FEB

BY DANNY MA

AGENDA:

- Intro (10 mins)
- Table Joins (50 mins)

WHAT WE COVERED LAST WEEK

- Marketing Analytics Case Study
- 5 W's (who/what/where/when/why)
- Campaign/business requirements
- Data exploration & ERD analysis
- Matching requirements to data

DVD RENTAL CO

Personalized recommendations from our very own team of film afficionados!

TRAVEL

You've watched 10 Travel films, that's 4 more than the DVD Rental Co average and puts you in the top 10% of Travel Gurus!

Your expertly chosen recommendations:
[Film #1 title goes here]
[Film #2 title goes here]
[Film #3 title goes here]

SCI-FI

You've watched 6 Sci-Fi films, making up 25% of your entire viewing history!

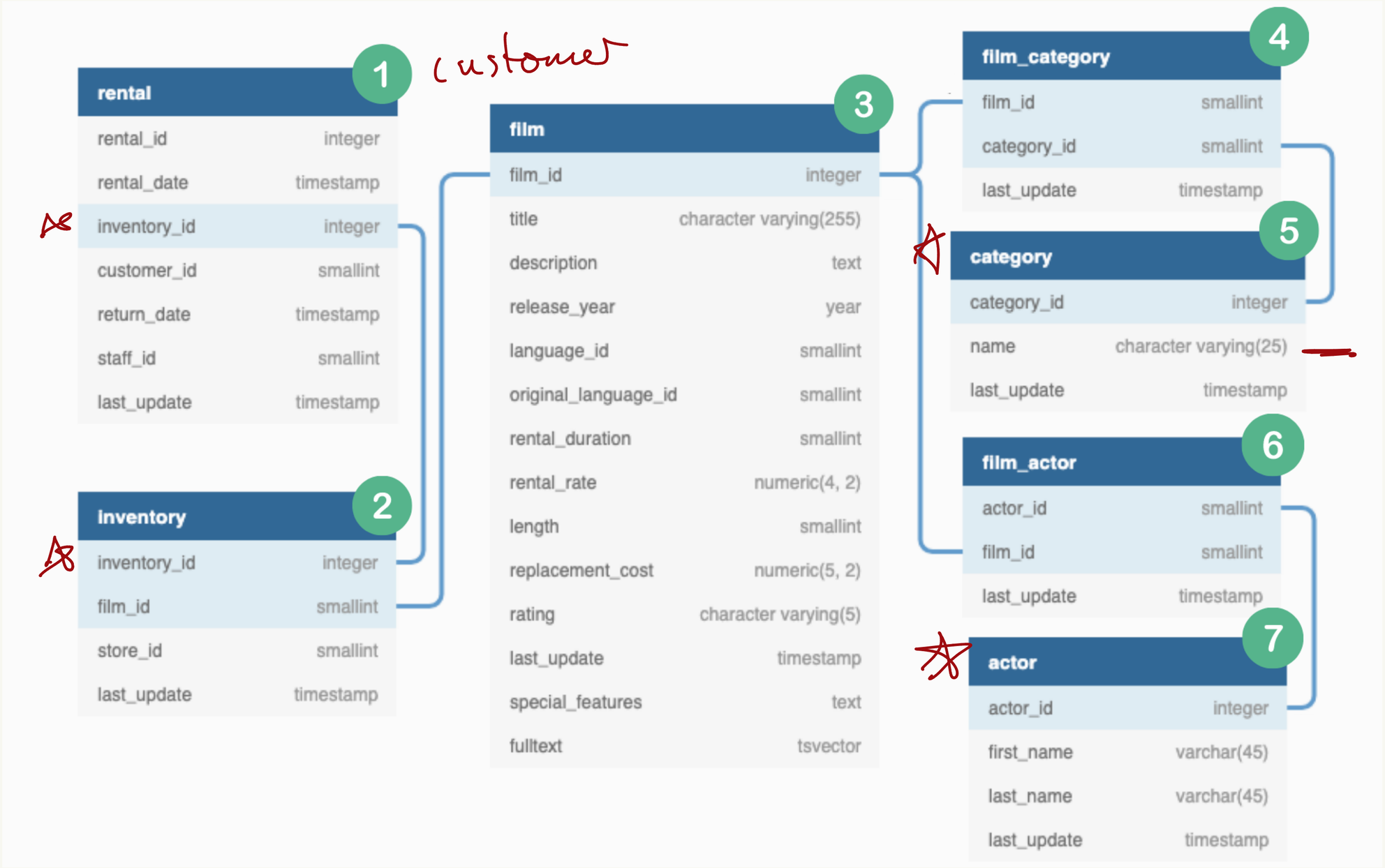
Your hand-picked recommendations:
[Film #1 title goes here]
[Film #2 title goes here]
[Film #3 title goes here]

RUFUS PAWS

You've watched 5 films featuring Rufus Paws! Here are some other films Rufus stars in that might interest you!

[Film #1 title goes here]
[Film #2 title goes here]
[Film #3 title goes here]

ENTITY RELATIONSHIP DIAGRAM



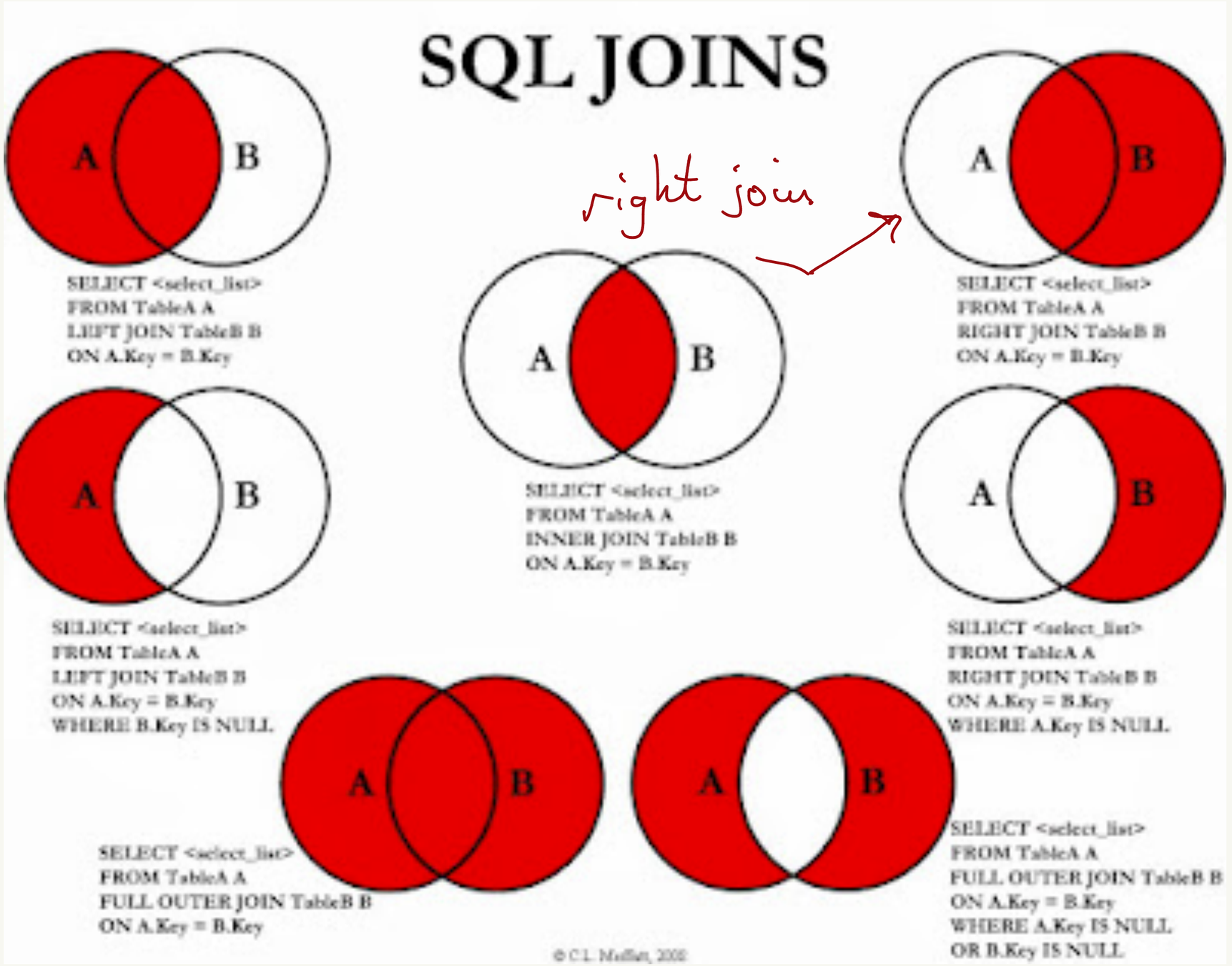
SQL TABLES JOINS

- INNER JOIN → JOIN
- LEFT JOIN → LEFT OUTER JOIN
- FULL JOIN → FULL OUTER JOIN
- CROSS JOIN → FROM Table1, Table2

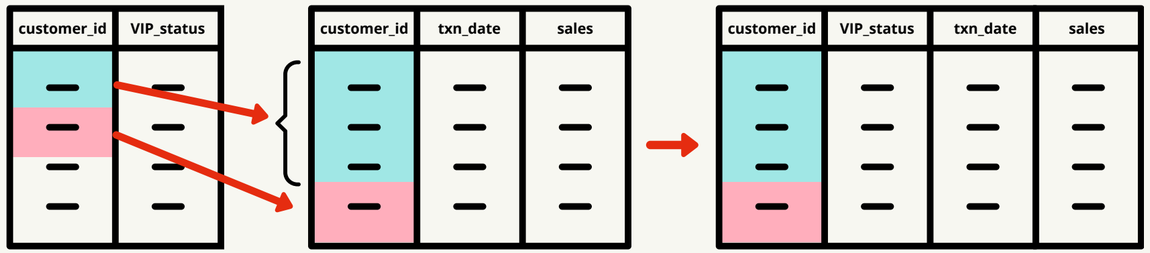
Advanced Joins

- LEFT SEMI JOIN
- ANTI JOIN

DO NOT LEARN TABLE JOINS LIKE THIS!

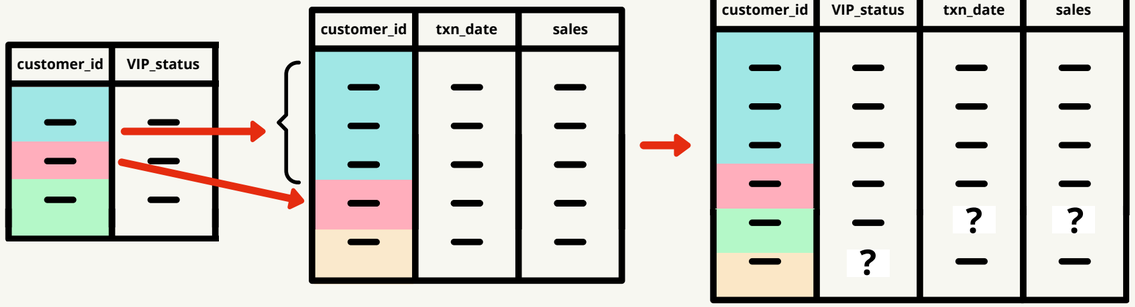


INNER JOIN



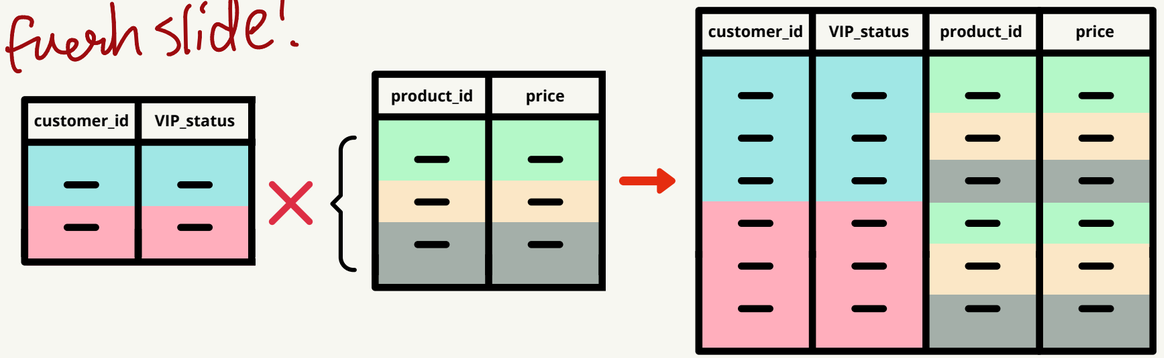
LEFT JOIN

FULL JOIN

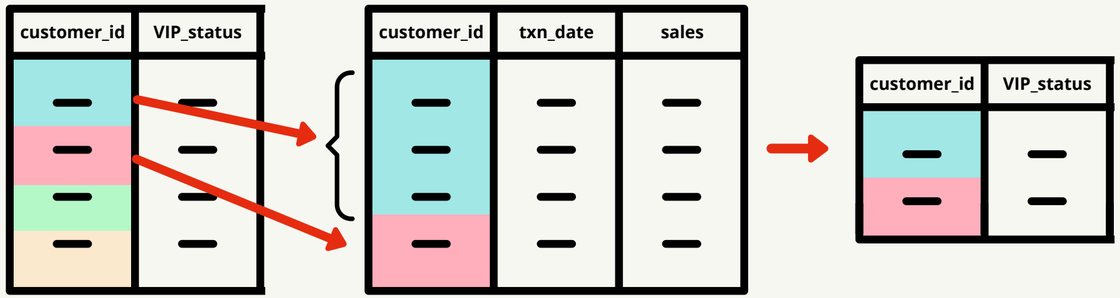


See new fresh slide!

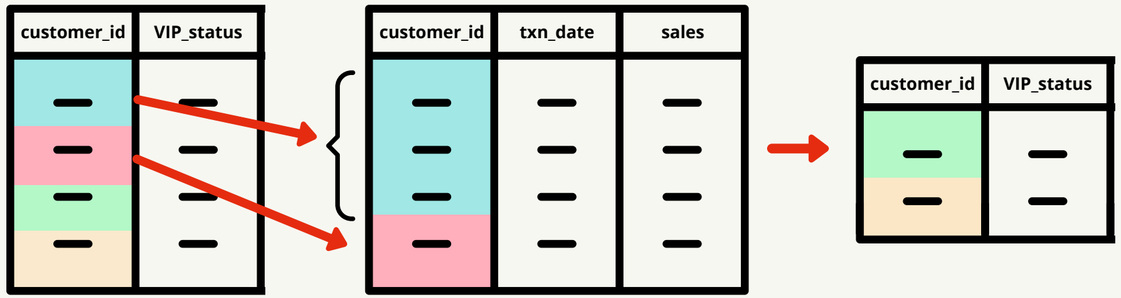
CROSS JOIN



LEFT SEMI JOIN



ANTI JOIN



LINKEDIN-FLUENCER DATA

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

jobs

iid	occupation	salary
1	Cleaner	High
2	Janitor	Medium
3	Monkey	Low
6	Plumber	Ultra
7	Hero	Plus Ultra

CREATE DATA SQL QUERY

```sql
DROP TABLE IF EXISTS names;
CREATE TEMP TABLE names AS
WITH input_data (iid, first_name, title) AS (
 VALUES
 (1, 'Kate', 'Datacated Visualizer'),
 (2, 'Eric', 'Captain SQL'),
 (3, 'Danny', 'Data Wizard Of Oz'),
 (4, 'Ben', 'Mad Scientist'),
 (5, 'Dave', 'Analytics Heretic'),
 (6, 'Ken', 'The YouTuber')
)
SELECT * FROM input_data;

DROP TABLE IF EXISTS jobs;
CREATE TEMP TABLE jobs AS
WITH input_data (iid, occupation, salary) AS (
 VALUES
 (1, 'Cleaner', 'High'),
 (2, 'Janitor', 'Medium'),
 (3, 'Monkey', 'Low'),
 (6, 'Plumber', 'Ultra'),
 (7, 'Hero', 'Plus Ultra')
)
SELECT * FROM input_data;
```

names of colums

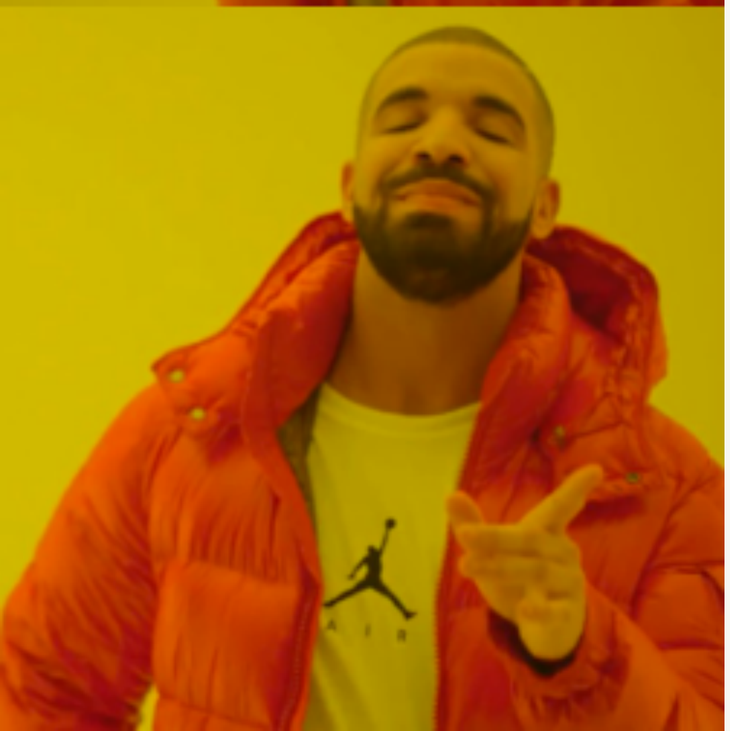
CTE

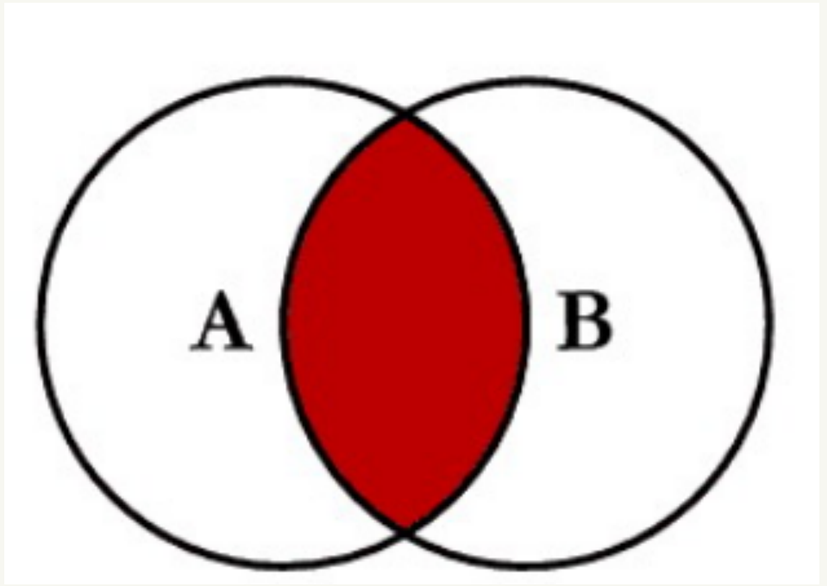
CTE method creating tables

replace w/ column names querying the CTE

select * from names

jobs





INNER JOIN

base table (left)

target table (right)

all 3 columns available

col1 col2

col1 col2 col3

customer_id	VIP_status
—	—
—	—
—	—
—	—

customer_id	txn_date	sales
—	—	—
—	—	—
—	—	—
—	—	—

customer_id	VIP_status	txn_date	sales
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—

BASIC INNER JOIN

Don't use * in joins
- duplicate columns
- lose source of columns

```
SELECT                          ← always use references
  names.iid,                    ← select expressions
  names.first_name,
  names.title,
  jobs.occupation,
  jobs.salary
FROM names                      ← base
INNER JOIN jobs                 ← target   (JOIN)
  ON names.iid = jobs.iid;      ← join construct
                                   (condition)
```

base

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	~~Ben~~	~~Mad Scientist~~
5	~~Dave~~	~~Analytics Heretic~~
6	Ken	The YouTuber

target

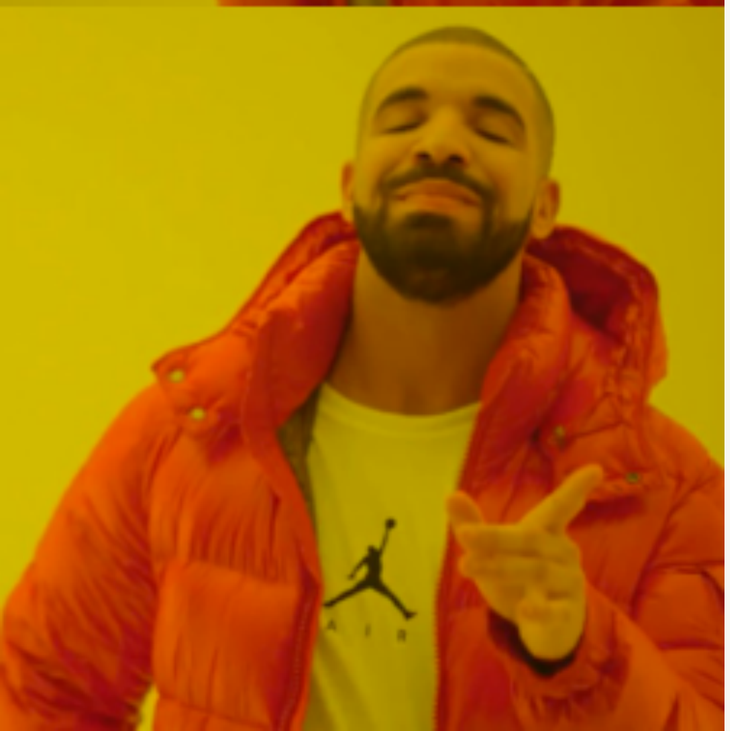
jobs

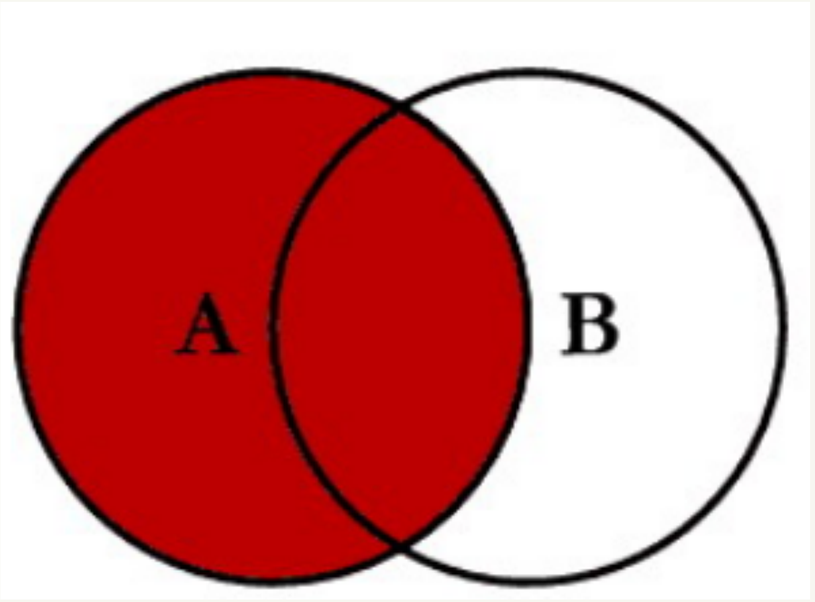
iid	occupation	salary
1	Cleaner	High
2	Janitor	Medium
3	Monkey	Low
6	Plumber	Ultra
~~7~~	~~Hero~~	~~Plus Ultra~~

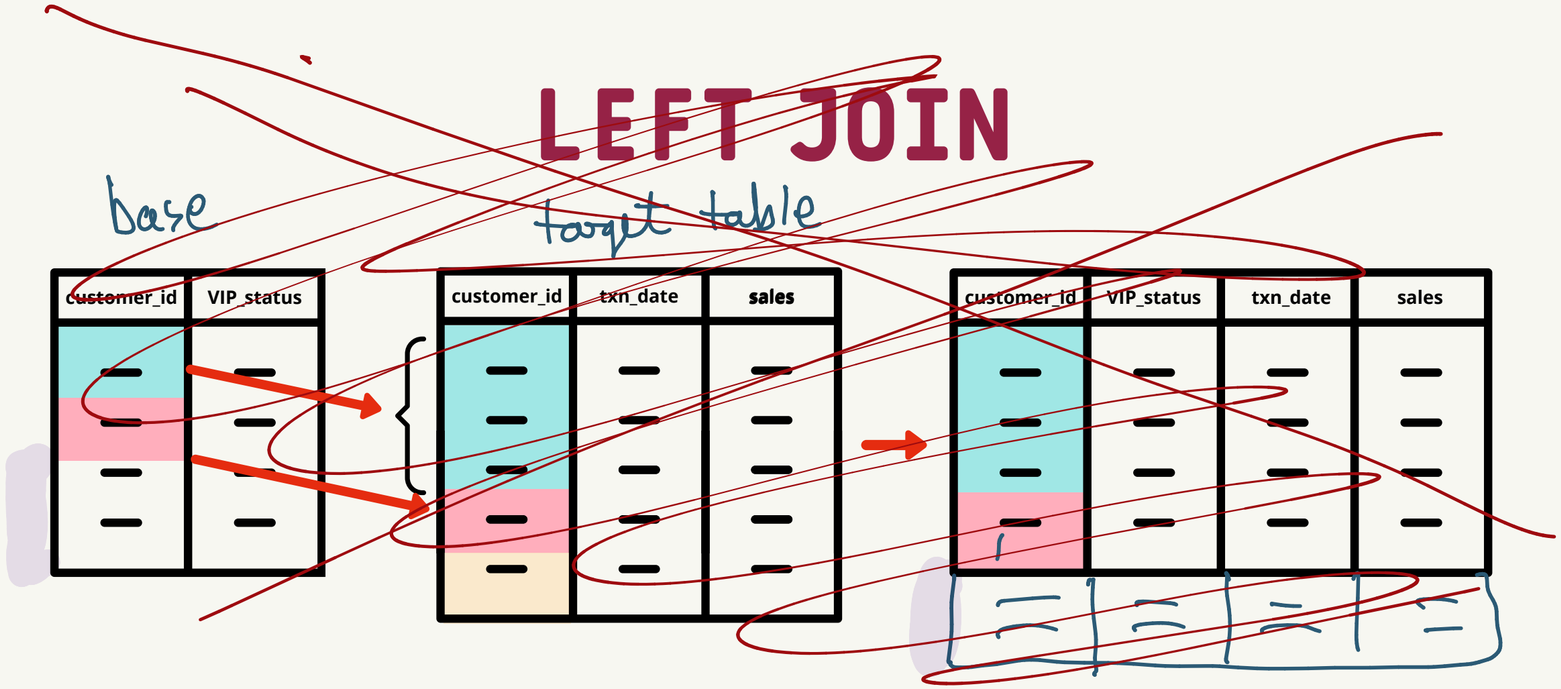
don't exist in target

doesn't exist in base

iid	first_name	title	occupation	salary
1	Kate	Datacated Visualizer	Cleaner	High
2	Eric	Captain SQL	Janitor	Medium
3	Danny	Data Wizard Of Oz	Monkey	Low
6	Ken	The YouTuber	Plumber	Ultra




A
B


LEFT JOIN
base
target table
customer_id
VIP_status
customer_id
txn_date
sales
customer_id
VIP_status
txn_date
sales

BASIC LEFT JOIN

```sql
SELECT
  names.iid,
  names.first_name,
  names.title,
  jobs.occupation,
  jobs.salary
FROM names
LEFT JOIN jobs
  ON names.iid = jobs.iid;
```

left outer join

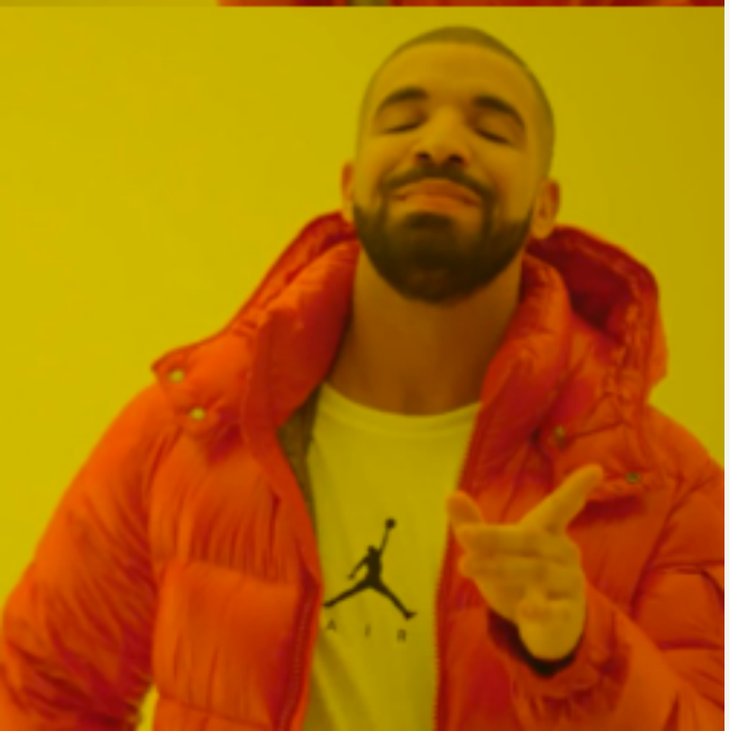
names

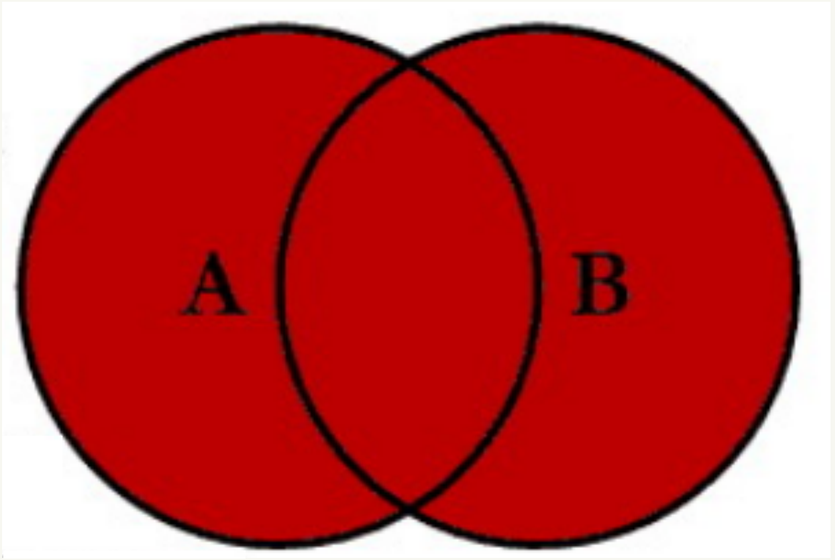
iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

jobs

iid	occupation	salary
1	Cleaner	High
2	Janitor	Medium
3	Monkey	Low
6	Plumber	Ultra
7	Hero	Plus Ultra

iid	first_name	title	occupation	salary
1	Kate	Datacated Visualizer	Cleaner	High
2	Eric	Captain SQL	Janitor	Medium
3	Danny	Data Wizard Of Oz	Monkey	Low
4	Ben	Mad Scientist	null	null
5	Dave	Analytics Heretic	null	null
6	Ken	The YouTuber	Plumber	Ultra





FULL JOIN

2

customer_id	VIP_status
—	—
—	—
—	—

3

customer_id	txn_date	sales
—	—	—
—	—	—
—	—	—
—	—	—
—	—	—

base table

target table

customer_id	VIP_status	txn_date	sales
—	—	—	—
—	—	—	—
—	—	—	—
—	—	—	—
—	—	?	?
—	?	—	—

BASIC FULL JOIN

```sql
SELECT
  names.iid AS name_id,
  jobs.iid AS job_id,
  names.first_name,
  names.title,
  jobs.occupation,
  jobs.salary
FROM names
FULL JOIN jobs
  ON names.iid = jobs.iid;
```

full outer

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

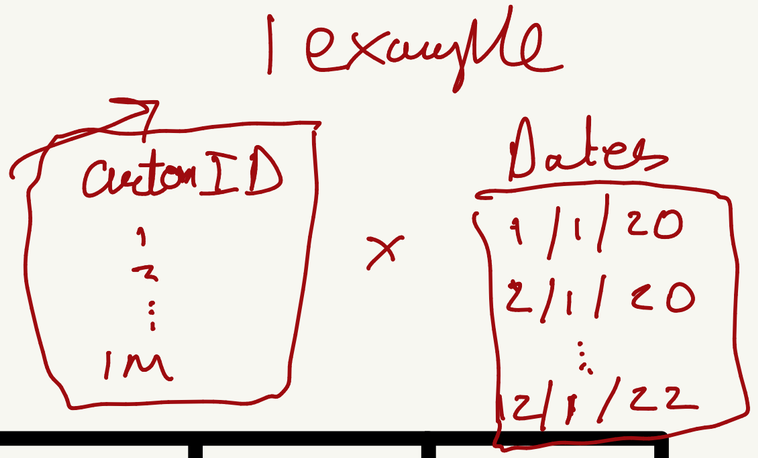
jobs

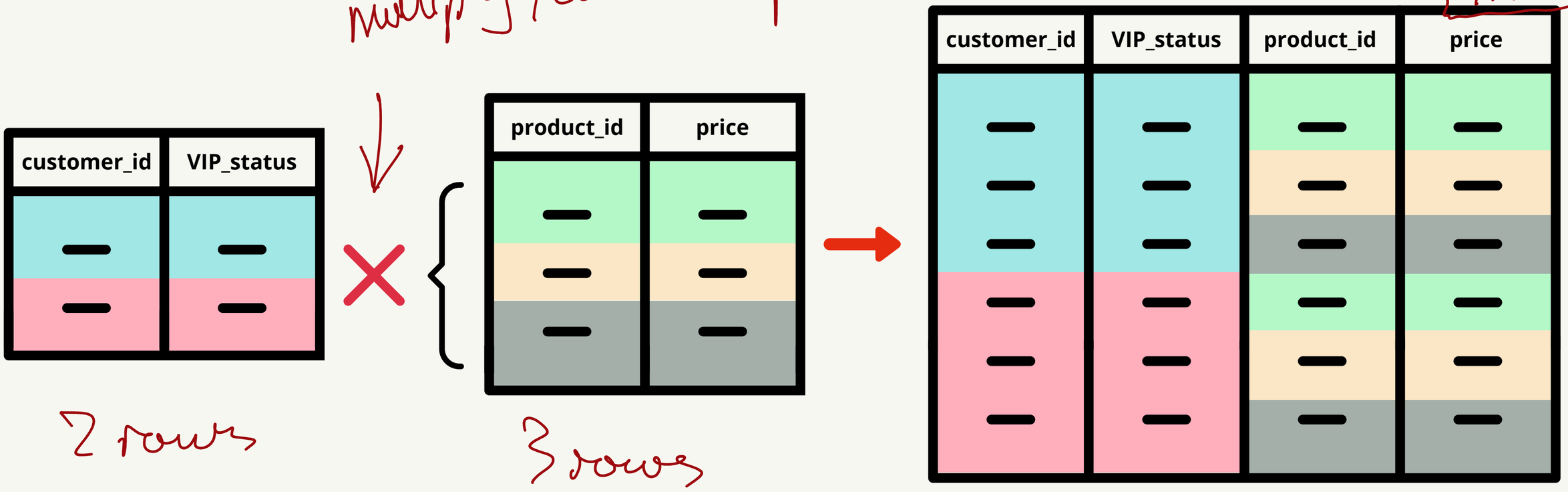
iid	occupation	salary
1	Cleaner	High
2	Janitor	Medium
3	Monkey	Low
6	Plumber	Ultra
7	Hero	Plus Ultra

name_id	job_id	first_name	title	occupation	salary
1	1	Kate	Datacated Visualizer	Cleaner	High
2	2	Eric	Captain SQL	Janitor	Medium
3	3	Danny	Data Wizard Of Oz	Monkey	Low
4	null	Ben	Mad Scientist	null	null
5	null	Dave	Analytics Heretic	null	null
6	6	Ken	The YouTuber	Plumber	Ultra
null	7	null	null	Hero	Plus Ultra

CROSS JOIN

multiply / cartesian product





BASIC CROSS JOIN

base
target

```sql
SELECT
    names.iid as name_iid,
    jobs.iid as job_iid,
    names.first_name,
    names.title,
    jobs.occupation,
    jobs.salary
FROM names
CROSS JOIN jobs;
```

FROM names, jobs
(Not recommended)

there's no ON condition

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

jobs

iid	occupation	salary
1	Cleaner	High
2	Janitor	Medium
3	Monkey	Low
6	Plumber	Ultra
7	Hero	Plus Ultra

name_iid	job_iid	first_name	title	occupation	salary
1	1	Kate	Datacated Visualizer	Cleaner	High
1	2	Kate	Datacated Visualizer	Janitor	Medium
1	3	Kate	Datacated Visualizer	Monkey	Low
1	6	Kate	Datacated Visualizer	Plumber	Ultra
1	7	Kate	Datacated Visualizer	Hero	Plus Ultra
2	1	Eric	Captain SQL	Cleaner	High
2	2	Eric	Captain SQL	Janitor	Medium
2	3	Eric	Captain SQL	Monkey	Low
2	6	Eric	Captain SQL	Plumber	Ultra
2	7	Eric	Captain SQL	Hero	Plus Ultra
3	1	Danny	Data Wizard Of Oz	Cleaner	High
3	2	Danny	Data Wizard Of Oz	Janitor	Medium
3	3	Danny	Data Wizard Of Oz	Monkey	Low
3	6	Danny	Data Wizard Of Oz	Plumber	Ultra
3	7	Danny	Data Wizard Of Oz	Hero	Plus Ultra
4	1	Ben	Mad Scientist	Cleaner	High
4	2	Ben	Mad Scientist	Janitor	Medium
4	3	Ben	Mad Scientist	Monkey	Low
4	6	Ben	Mad Scientist	Plumber	Ultra
4	7	Ben	Mad Scientist	Hero	Plus Ultra
5	1	Dave	Analytics Heretic	Cleaner	High
5	2	Dave	Analytics Heretic	Janitor	Medium
5	3	Dave	Analytics Heretic	Monkey	Low
5	6	Dave	Analytics Heretic	Plumber	Ultra
5	7	Dave	Analytics Heretic	Hero	Plus Ultra
6	1	Ken	The YouTuber	Cleaner	High
6	2	Ken	The YouTuber	Janitor	Medium
6	3	Ken	The YouTuber	Monkey	Low
6	6	Ken	The YouTuber	Plumber	Ultra
6	7	Ken	The YouTuber	Hero	Plus Ultra

TABLES WITH DUPLICATES

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

new_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra

DUPLICATE INNER JOIN

```sql
SELECT
  names.iid,
  names.first_name,
  names.title,
  new_jobs.occupation,
  new_jobs.salary
FROM names
INNER JOIN new_jobs
  ON names.iid = new_jobs.iid;
```

reference

duplicate

base

target

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

new_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra

iid	first_name	title	occupation	salary
1	Kate	Datacated Visualizer	Cleaner	Very High
1	Kate	Datacated Visualizer	Cleaner	High
2	Eric	Captain SQL	Janitor	Medium
3	Danny	Data Wizard Of Oz	Monkey	Very Low
3	Danny	Data Wizard Of Oz	Monkey	Low
6	Ken	The YouTuber	Plumber	Ultra

DUPLICATE LEFT JOIN

```sql
SELECT
  names.iid,
  names.first_name,
  names.title,
  new_jobs.occupation,
  new_jobs.salary
FROM names
LEFT JOIN new_jobs
  ON names.iid = new_jobs.iid;
```

names

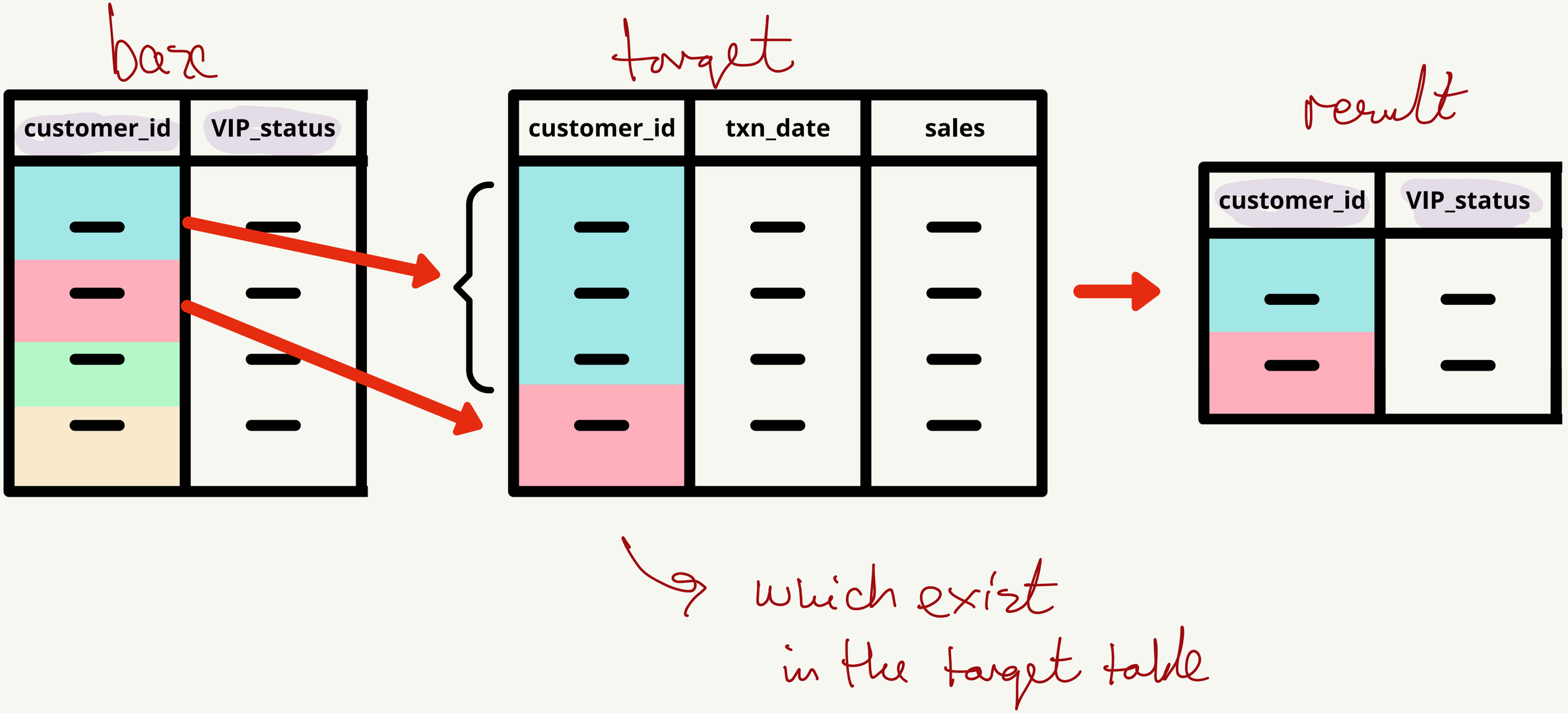
iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

new_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra

iid	first_name	title	occupation	salary
1	Kate	Datacated Visualizer	Cleaner	Very High
1	Kate	Datacated Visualizer	Cleaner	High
2	Eric	Captain SQL	Janitor	Medium
3	Danny	Data Wizard Of Oz	Monkey	Very Low
3	Danny	Data Wizard Of Oz	Monkey	Low
4	Ben	Mad Scientist	null	null
5	Dave	Analytics Heretic	null	null
6	Ken	The YouTuber	Plumber	Ultra

LEFT SEMI JOIN



LEFT SEMI JOIN

```sql
SELECT
  names.iid,
  names.first_name
FROM names
WHERE EXISTS (
  SELECT 1
  FROM new_jobs
  WHERE names.iid = new_jobs.iid
);
```

Postgres specific

add a comment

correlated subquery

does not return any data

ON condition

```sql
SELECT
  names.iid,
  names.first_name
FROM names
LEFT SEMI JOIN new_jobs
ON names.iid = new_jobs.iid;
```

won't run in postgres

base

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
~~4~~	Ben	Mad Scientist
~~5~~	Dave	Analytics Heretic
6	Ken	The YouTuber

target

new_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra

iid	first_name
1	Kate
2	Eric
3	Danny
6	Ken

LEFT SEMI JOIN OR LEFT JOIN WITHOUT NULLS?

```sql
SELECT DISTINCT
  names.iid,
  names.first_name
FROM names
LEFT JOIN new_jobs
  ON names.iid = new_jobs.iid
WHERE new_jobs.iid IS NOT NULL;
```

→ removed

filter out / remove rows

target table

base

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

new_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra

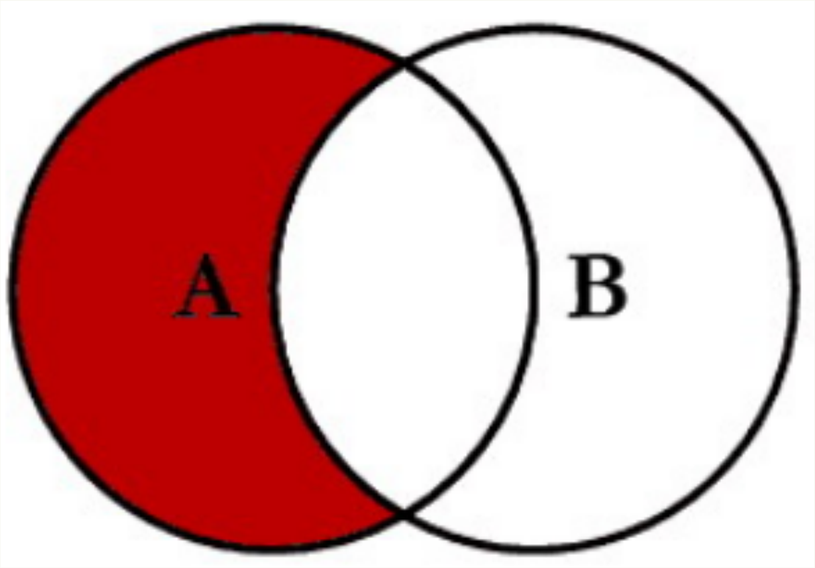
SCD

DISTINCT

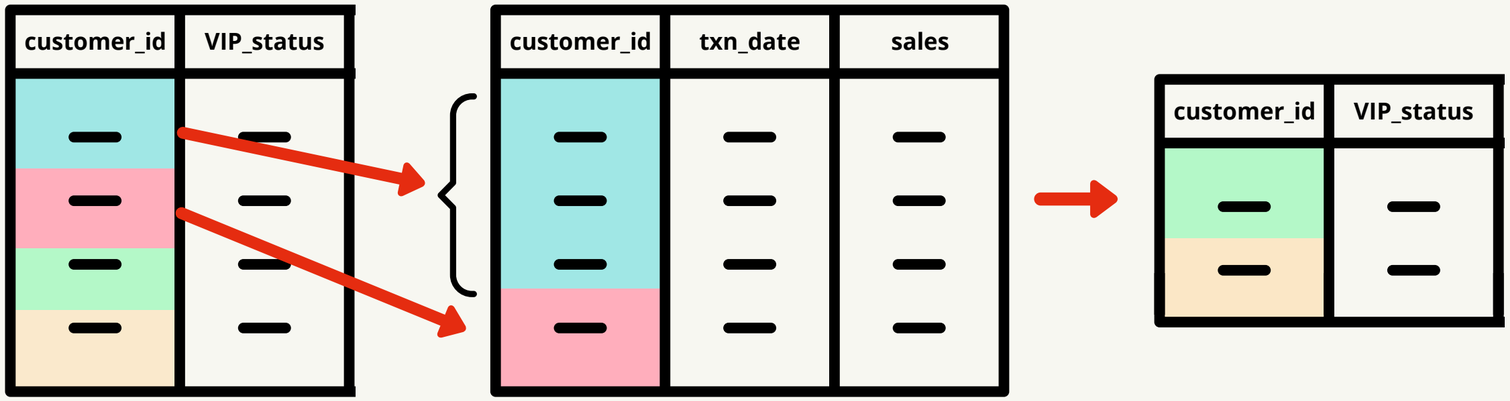
iid	first_name
1	Kate
2	Eric
3	Danny
6	Ken

w/o Distinct

iid	first_name
1	Kate
1	Kate
2	Eric
3	Danny
3	Danny
6	Ken


A
B

ANTI JOIN


customer_id
VIP_status
customer_id
txn_date
sales
customer_id
VIP_status

ANTI JOIN

```sql
SELECT
  names.iid,
  names.first_name
FROM names
WHERE NOT EXISTS (
  SELECT 1
  FROM new_jobs
  WHERE names.iid = new_jobs.iid
);
```

```sql
SELECT
  names.iid,
  names.first_name
FROM names
ANTI JOIN new_jobs
  ON names.iid = new_jobs.iid
```

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

new_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra

iid	first_name
4	Ben
5	Dave

ANTI JOIN OR LEFT JOIN WITH NULL FILTER?

```sql
SELECT
  names.iid,
  names.first_name
FROM names
LEFT JOIN new_jobs
  ON names.iid = new_jobs.iid
WHERE new_jobs.iid IS NULL;
```

new_jobs.iid

NO NOT

names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber

new_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra

iid	first_name	title	occupation	salary
1	Kate	Datacated Visualizer	Cleaner	Very High
1	Kate	Datacated Visualizer	Cleaner	High
2	Eric	Captain SQL	Janitor	Medium
3	Danny	Data Wizard Of Oz	Monkey	Very Low
3	Danny	Data Wizard Of Oz	Monkey	Low
4	Ben	Mad Scientist	null	null
5	Dave	Analytics Heretic	null	null
6	Ken	The YouTuber	Plumber	Ultra

iid	first_name
4	Ben
5	Dave

DEALING WITH NULL VALUES WILL IT JOIN???

null_names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber
null	Giorno	OG Data Gangster

null_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra
null	Mastermind	Bank

INNER JOIN?

Does not join w/ null values

```
SELECT
  names.iid,
  names.first_name,
  names.title,
  jobs.occupation,
  jobs.salary
FROM null_names AS names
INNER JOIN null_jobs AS jobs
  ON names.iid = jobs.iid;
```

null_names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber
null	Giorno	OG Data Gangster

null_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra
null	Mastermind	Bank

iid	first_name	title	occupation	salary
1	Kate	Datacated Visualizer	Cleaner	Very High
1	Kate	Datacated Visualizer	Cleaner	High
2	Eric	Captain SQL	Janitor	Medium
3	Danny	Data Wizard Of Oz	Monkey	Very Low
3	Danny	Data Wizard Of Oz	Monkey	Low
6	Ken	The YouTuber	Plumber	Ultra

=> ? > ?

LEFT JOIN?

Does not join

```
SELECT
  names.iid,
  names.first_name,
  names.title,
  jobs.occupation,
  jobs.salary
FROM null_names AS names
LEFT JOIN null_jobs AS jobs
  ON names.iid = jobs.iid;
```

null_names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber
null	Giorno	OG Data Gangster

null_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra
null	Mastermind	Bank

target

iid	first_name	title	occupation	salary
1	Kate	Datacated Visualizer	Cleaner	Very High
1	Kate	Datacated Visualizer	Cleaner	High
2	Eric	Captain SQL	Janitor	Medium
3	Danny	Data Wizard Of Oz	Monkey	Very Low
3	Danny	Data Wizard Of Oz	Monkey	Low
4	Ben	Mad Scientist	null	null
5	Dave	Analytics Heretic	null	null
6	Ken	The YouTuber	Plumber	Ultra
null	Giorno	OG Data Gangster	null ???	null ???

LEFT SEMI JOIN?

↳ Doesn't join

```
SELECT
  null_names.* ← naughty
FROM null_names ← only columns from bare table
WHERE EXISTS (
  SELECT iid
  FROM null_jobs
  WHERE null_names.iid = null_jobs.iid
);
```

null_names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber
null	Giorno	OG Data Gangster

null_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra
null	Mastermind	Bank

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
6	Ken	The YouTuber

ANTI JOIN?

(works???. kind of

```sql
SELECT
  null_names.*
FROM null_names
WHERE NOT EXISTS (
  SELECT iid
  FROM null_jobs
  WHERE null_names.iid = null_jobs.iid
);
```

null_names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber
null	Giorno	OG Data Gangster

null_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra
null	Mastermind	Bank

iid	first_name	title
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
null	Giorno	OG Data Gangster

WHAT ABOUT IN INSTEAD OF WHERE EXISTS?

```sql
SELECT
  null_names.*
FROM null_names
WHERE EXISTS (
  SELECT iid
  FROM null_jobs
  WHERE null_names.iid = null_jobs.iid
);
```

← can also be 1 doesn't matter

```sql
SELECT
  null_names.*
FROM null_names
WHERE null_names.iid IN (
  SELECT iid
  FROM null_jobs
);
```

← returns data

IN pattern

null_names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber
null	Giorno	OG Data Gangster

null_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra
null	Mastermind	Bank

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
6	Ken	The YouTuber

NOT IN INSTEAD OF ANTI JOIN?

null_names

iid	first_name	title
1	Kate	Datacated Visualizer
2	Eric	Captain SQL
3	Danny	Data Wizard Of Oz
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
6	Ken	The YouTuber
null	Giorno	OG Data Gangster

null_jobs

iid	occupation	salary
1	Cleaner	High
1	Cleaner	Very High
2	Janitor	Medium
3	Monkey	Low
3	Monkey	Very Low
6	Plumber	Ultra
7	Hero	Plus Ultra
null	Mastermind	Bank

```sql
SELECT
  null_names.*
FROM null_names
WHERE NOT EXISTS (
  SELECT iid
  FROM null_jobs
  WHERE null_names.iid = null_jobs.iid
);
```

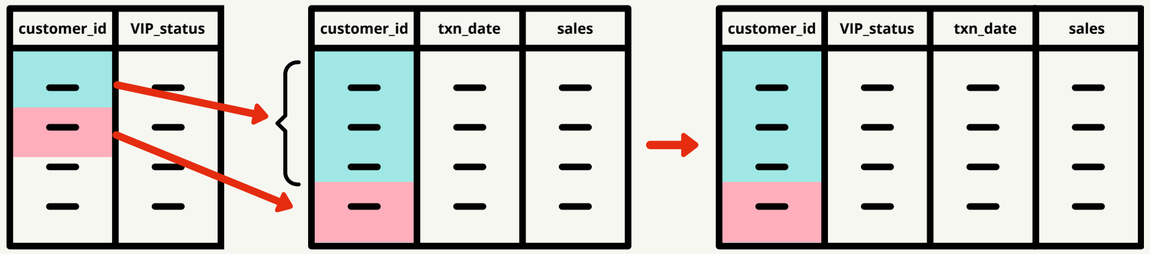
iid	first_name	title
4	Ben	Mad Scientist
5	Dave	Analytics Heretic
null	Giorno	OG Data Gangster

```sql
SELECT
  null_names.*
FROM null_names
WHERE null_names.iid NOT IN (
  SELECT
    iid
  FROM null_jobs
);
```

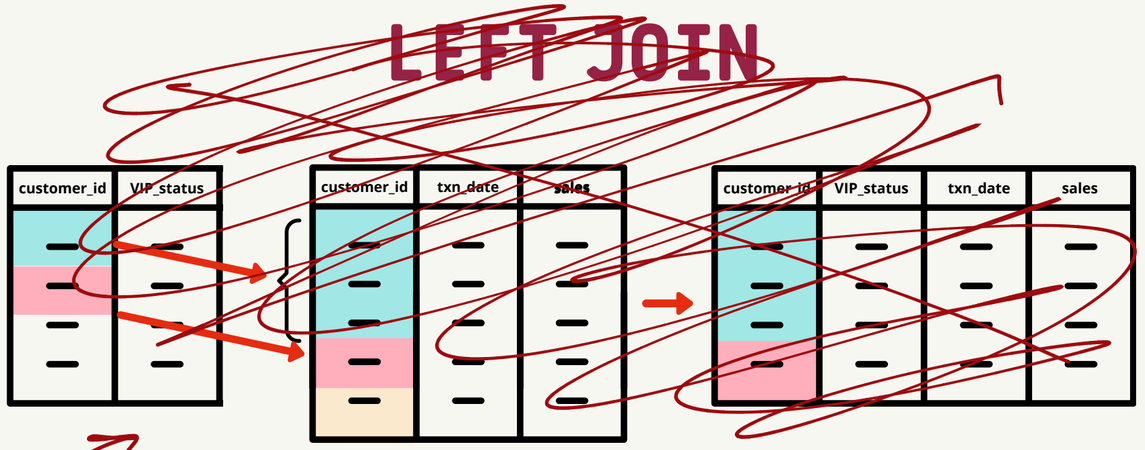
iid	first_name	title

Why is this blank

INNER JOIN

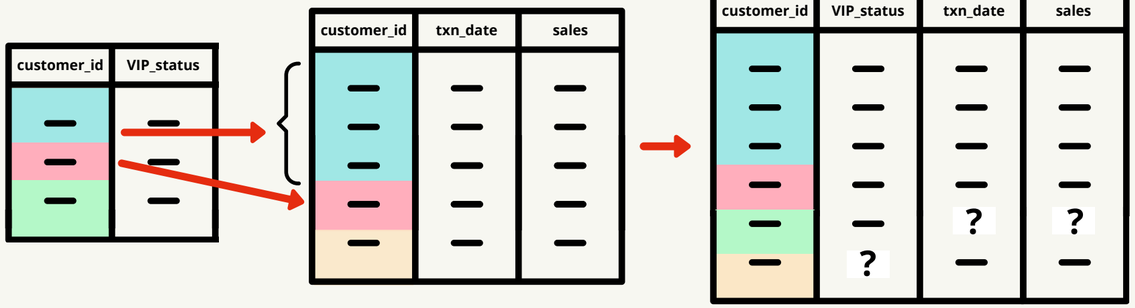


LEFT JOIN

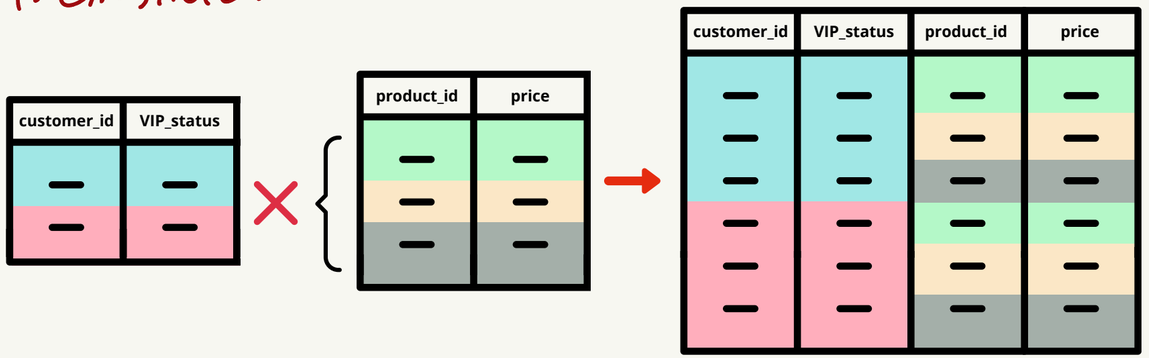


See new freshslides!

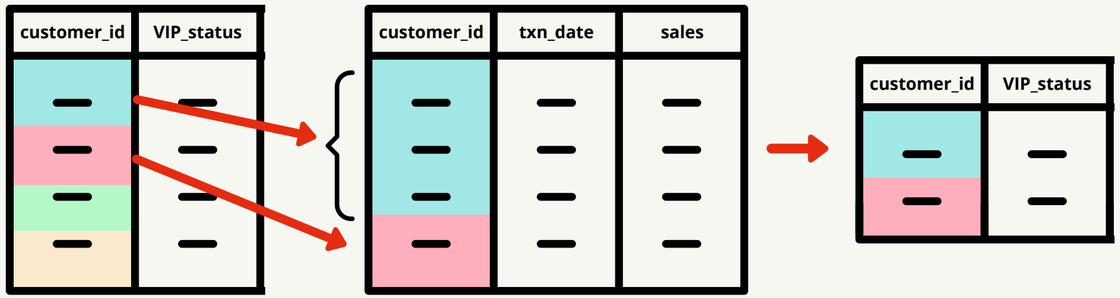
FULL JOIN



CROSS JOIN



LEFT SEMI JOIN



ANTI JOIN