Introduction to Data Structure (Data Management)

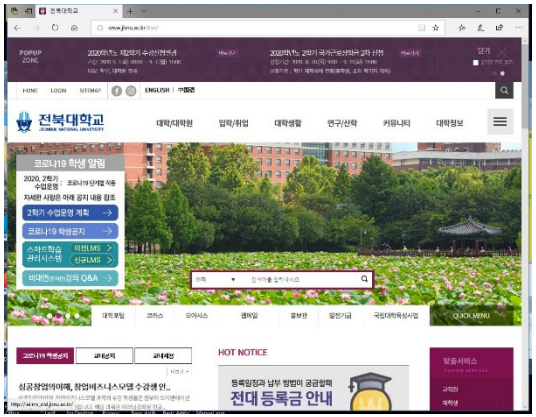
Felipe P. Vista IV

Reminder

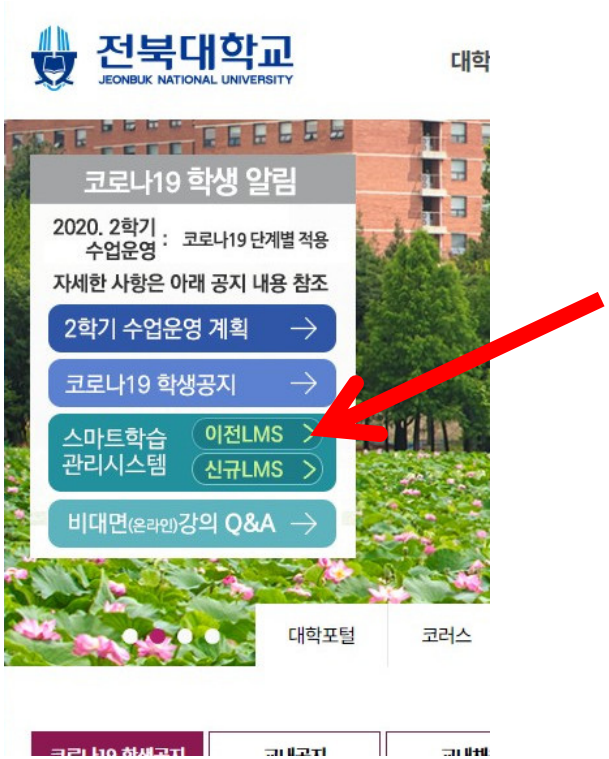
- Everybody, make sure that your name in ZOOM is in the following format:
  - University ID Num Name
  - Ex: 202054321 Juan Dela Cruz
  - Everybody, prepare to turn on your webcam by 2:10pm
  - Not changing your name to this format
    - you might be marked Absent

    * Farrukhbek → absent?

Announcement



http://jbnu.ac.kr/



http://ieilms_old.jbnu.ac.kr/ → http://ieilmsold.jbnu.ac.kr/

Course Outline

- Introduction, Data Models, SQL Basics
- SQL Aggregates, Grouping, Subqueries
- Wrapping-up SQL, Relational Algebra (RA), Datalog
- NoSQL, JSON
- JSON, SQL++
- SQL++, RA Part II, Query Evaluation
- Storage, Indexing Basics
- Basics of Query Optimization, Parallel Databases
- Map Reduce, Spark
- E/R Diagrams, Constraints
- Design Theory
- Transactions
- DB Techniques for Machine Learning

Data Management

JOINS & AGGREGATION

Joins and Aggregation

- Inner Joins (6.2)
- Outer Joins (6.3.8)
- Aggregations (6.4.3 – 6.4.6)

Unique

- PRIMARY KEY adds implicit "NOT NULL" constraint while UNIQUE does not
  - Add 'UNIQUE' implicitly

Primary key = 1
Unique keys ≥ 1

```
CREATE TABLE Crew(
    – name VARCHAR(20) NOT NULL, …
      UNIQUE (name));
```

Unique

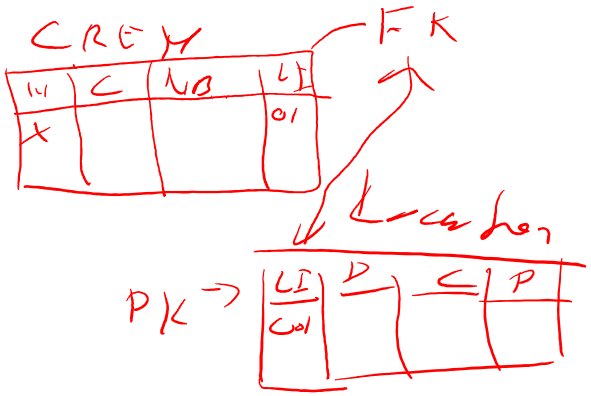
- PRIMARY KEY adds implicit "NOT NULL" constraint while UNIQUE does not
  - Add 'UNIQUE' implicitly

name	AGE
~~NULL~~	20
NULL	

```
CREATE TABLE Crew(
        name VARCHAR(20) NOT NULL, …
        UNIQUE (name));
```

- Need to do this
  - SQL Server will behave strangely will NULL & UNIQUE
  - always consider about NULL when querying
  - remove NOT NULL constraint if you want in the future

Inner Joins




```
SELECT f01, f02, ..., fn
   FROM T01, T02, ..., Tn
WHERE condition(s);
```

Inner Joins

```
SELECT f01, f02, ..., fn
   FROM T01, T02, ..., Tn
WHERE condition(s);
```

→ fields/cols
→ Tables

- Seeing it as a nested loop:

```
for x1 in T1:
   for x2 in T2:
      …
         for xn in Tn:
            if condition(x1.f01, x2.f02, …):
               output(x1.f01, x1.f02, …, xn.fn)
```

T01.f01 = T02.F01

Inner Joins

company(cname, country)

product(pname, price, category, manuf)

- "manuf" is FOREIGN KEY to table crew company

```
CREATE TABLE company(
        cname VARCHAR(40), country VARCHAR(40),
        PRIMARY KEY (cname));
CREATE TABLE product(
        pname VARCHAR(40), price DECIMAL(6,2),
        category VARCHAR(30),
        manuf VARCHAR(40),
        FOREIGN KEY (manuf)
           REFERENCES company(cname));
```

Inner Joins

company(cname, country)

product(pname, price, category, manuf)

- "manuf" is FOREIGN KEY to table crew

```
SELECT DISTINCT cname
   FROM product, company
WHERE country = 'South Korea' AND category = 'car' AND
   manuf = cname;
```

Inner Joins (suggested habit)

company(cname, country)

product(pname, price, category, cname)

FK → COT

manuf

- "cname" is FOREIGN KEY to table crew

```
CREATE TABLE company(
      cname VARCHAR(40), country VARCHAR(40),
      PRIMARY KEY (cname));
CREATE TABLE product(
      pname VARCHAR(40), price DECIMAL(6,2),
      category VARCHAR(30),
      cname VARCHAR(40),
      FOREIGN KEY (cname)
         REFERENCES company(cname));
```

Inner Joins (suggested habit)

company(cname, country)

product(pname, price, category, cname)

- "cname" is FOREIGN KEY to table crew

T F

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='south korea' AND
   product.category='car' AND product.cname=company.cname;
```

1 = FK
2 = PK

1 2

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

lower

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

pname	cateegory	cname	cname	country
genesis	car	hyundai	hyundai	south korea

(1) →

?. → How many?

4/2/3/4
/5 ...

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

No output since company.country('italy') != 'south korea'
Also since product.cname('hyundai') != company.cname('ducati')

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

No output since company.country('italy') != 'south korea'
Also since product.cname('hyundai') != company.cname('vespa')

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

No output since product.category('motorcycle') != 'car'
Also product.cname('ducati') != company.cname('hyundai')

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
Hyundai	South Korea
Ducati	Italy
Vespa	Italy

No output since product.category('motorcycle') != 'car'
Also since company.country('italy') != 'south krea'

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

No output since product.category('motorcycle') != 'car'
Also since company.country('italy') != 'south korea'
Also product.cname('ducati') != company.cname('vespa')

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

No output since product.category('motorcycle') != 'car'
Also since product.cname('vespa') != company.cname('hyundai')

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

No output since company.country('italy') != 'south korea'
Also product.category('motorcycle') != 'car'
Also since product.cname('vespa') != company.cname('ducati')

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

No output since company.country('italy') != 'south korea'
Also product.category('motorcycle') != 'car'

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Restrict product.category = 'car'

Limit

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
piaggio	motorcycle	vespa

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Restrict product.category = 'car'

Product (where category='car')

pname	category	cname
genesis	car	hyundai

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Restrict company.country = 'south korea'

Product (where category='car')

pname	category	cname
genesis	car	hyundai

Company

cname	country
hyundai	south korea
ducati	italy
vespa	italy

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Restrict company.country = 'south korea'

Product (where category='car')

pname	category	cname
genesis	car	hyundai

Company (where country='south korea')

cname	country
hyundai	south korea

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Restrict product.cname= company.cname

Only one combination to consider

Product (where category='car')

pname	category	cname
genesis	car	hyundai

Company (where country='south korea')

cname	country
hyundai	south korea

Inner Joins

```
SELECT DISTINCT company.cname
   FROM product, company
WHERE company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Alternative syntax:

```
SELECT DISTINCT company.cname
   FROM product JOIN company
ON company.country='South Korea' AND
   product.category='car' AND product.cname=company.cname;
```

Emphasize that the *predicate is part of the join

* predicate – condition expression that evaluates to a boolean value, either TRUE or FALSE

Self-Joins and Tuple Variable

- Ex: What companies manufacture products in the 'car' and 'motorcycle' categories
  - Just joining company with product is not enough; need to join company w/ product and another product

```
FROM company, product, product…
```

Self-Joins and Tuple Variable

- Ex: What companies manufacture products in the 'car' and 'motorcycle' categories
  - Just joining company with product is not enough; need to join company w/ product and another product

```
FROM company, product, product…
```

CAR
MOTORCYCLE
CAR
MOTOR

- a relationship that occurs twice in the FROM clause is called *self-join*
  - Every column name will be ambiguous if we run sql

* ambiguous – not clear, no choice between alternatives has been made

Name Conflicts

- If field name is ambiguous, qualify it:

From: `WHERE company.cname = product.pname…` – CNR

To: `WHERE company.cname = product.pname…` – MOTOR

Name Conflicts

- If field name is ambiguous, qualify it:

  From: `WHERE company.name = product.name…`

  To: `WHERE company.cname = product.pname…`

- For self-join, distinguish the tables:

  `FROM product X, product Y, company`

Name Conflicts

- If field name is ambiguous, qualify it:

  From: `WHERE company.name = product.name…`

  To: `WHERE company.cname = product.pname…`

- For self-join, distinguish the tables:

  `FROM product X, product Y, company Z, ...`

- These names are called "*tuple variables*"
  - imagine of as name for variable of each loop
  - can also write "`company AS Z`", etc…
  - can make the SQL statement shorter:
    - "Z.name" vs "company.name"

Self-joins

```
SELECT DISTINCT Z.cname
   FROM product X, product Y, company Z
WHERE Z.country='south korea'
   AND X.category='car' AND Y.category='motorcycle'
   AND X.cname=Z.cname AND Y.cname=Z.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
sattavis	motorcycle	hyundai

Company

cname	country
hyundai	south korea
ducati	italy

Self-joins

```
SELECT DISTINCT Z.cname
   FROM product X, product Y, company Z
WHERE Z.country='south korea'
   AND X.category='car' AND Y.category='motorcycle'
   AND X.cname=Z.cname AND Y.cname=Z.cname;
```

Product

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
sattavis	motorcycle	hyundai

Company

cname	country
hyundai	south korea
ducati	italy

No output since Y.category('car') != 'motorcycle'

Self-joins

```
SELECT DISTINCT Z.cname
   FROM product X, product Y, company Z
WHERE Z.country='south korea'
   AND X.category='car' AND Y.category='motorcycle'
   AND X.cname=Z.cname AND Y.cname=Z.cname;
```

Product

X

pname	category	cname
genesis	car	hyundai
multistrada	motorcycle	ducati
sattavis	motorcycle	hyundai

Y

Company

cname	country
hyundai	south korea
ducati	italy

Z

No output since Y.cname('ducati') != Z.cname('hyundai')

Self-joins

```
SELECT DISTINCT Z.cname
   FROM product X, product Y, company Z
WHERE Z.country='south korea'
   AND X.category='car' AND Y.category='motorcycle'
   AND X.cname=Z.cname AND Y.cname=Z.cname;
```

Product

	pname	category	cname
X	genesis	car	hyundai
	multistrada	motorcycle	ducati
Y	sattavis	motorcycle	hyundai

Company

cname	country	
hyundai	south korea	Z
ducati	italy	

X_pname	X_category	X_cname	Y_pname	Y_category	Y_cname	Z_cname	Z_country
genesis	car	hyundai	sattavis	motorcycle	hyundai	hyundai	south korea

Outer Joins

T_2 ✓ T_1

FK = PK

PK

product(name, category)

FK

purchase(prodName, store,…)

- "prodName" is FOREIGN KEY to table product (name)

```
SELECT product.name, …, product.store
   FROM product, purchase
WHERE product.name = purchase.prodName;
```

Outer Joins

product(name, category)

purchase(prodName, store,…)

- "prodName" is FOREIGN KEY to table product

```
SELECT product.name, …, product.store
   FROM product, purchase
WHERE product.name = purchase.prodName;
```

Or equivalently:

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

But some products may not be listed, why?

We will not be selecting products that were not sold ☹…

Outer Joins

```
product(name, category)
purchase(prodName, store,…)
   - "prodName" is FOREIGN KEY to table product
```

If we want to include products that were never sold, then we need an '*outer join*'

```
SELECT product.name, …, product.store
   FROM product LEFT OUTER JOIN purchase
ON product.name = purchase.prodName;
```

Outer Joins

- LEFT OUTER JOIN row/record
  - include the left tuple even if there is no match
- RIGHT OUTER JOIN
  - include the right tuple even if there is no match
- FULL OUTER JOIN
  - include both left and right tuples even if there is no match

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats
multistrada	nyavideo

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats
multistrada	nyavideo
multistrada	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats
multistrada	nyavideo
multistrada	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats
multistrada	nyavideo
multistrada	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats
multistrada	nyavideo
multistrada	patti-pats

Outer Joins

```
SELECT product.name, …, product.store
   FROM product LEFT OUTER JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

Outer Joins

```
SELECT product.name, …, product.store
   FROM product LEFT OUTER JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's

name	store
genesis	patti-pats
multistrada	nyavideo
multistrada	patti-pats
sattavis	NULL

Outer Joins

```
SELECT product.name, …, product.store
   FROM product RIGHT OUTER JOIN purchase
ON product.name = purchase.prodName;
```

LOJ

Product PK

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase FK

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's
pajero	nwabs

genesis - p
multi - n
multi - p
s

LOJ
+ sattavis - NULL

gen - p
mult - n
mult - p
s - NULL

QAW - JG

Outer Joins

```
SELECT product.name, …, product.store
   FROM product RIGHT OUTER JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's
pajero	nwabs

name	store
genesis	patti-pats
multistrada	nyavideo
multistrada	patti-pats
NULL	nwabs

Outer Joins

```
SELECT product.name, …, product.store
   FROM product FULL OUTER JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's
pajero	nwabs

OJ → 3 recs
LOJ → 4 recs
ROJ → 4 recs
FOJ → recs ?

Outer Joins

```
SELECT product.name, …, product.store
   FROM product FULL OUTER JOIN purchase
ON product.name = purchase.prodName;
```

Product

name	category
genesis	car
multistrada	motorcycle
sattavis	motorcycle

Purchase

prodName	store
genesis	patti-pat's
multistrada	nyavideo
multistrada	patti-pat's
pajero	nwabs

name	store
genesis	patti-pats
multistrada	nyavideo
multistrada	patti-pats
sattavis	NULL
NULL	nwabs

Aggregation in SQL

```
$>sqlite3 lecWk02
```

dBname

```
sqlite> create table purchase (
            pid int primary key,
            product text,
            price float,
            quantity int,
            month varchar(15));
```

1

```
sqlite> .import lecWk02-data.txt Purchase
```

2