

# Lecture 3: SQL

Last Day: Basic Syntax of Queries.

Select Attributes from Tables where Conditions	+ Frills
--	----------

Today: Frills, Queries, Updates, Views,  
Creating database tables.

See Korth & Silberschatz, § 4.1

# Basic Syntax: Example

2

## SUPPLIERS

NAME	ITEM	PRICE
Acme	Brie	2.49
Acme	Perrier	1.19
Acme	Snails	.25
Ajax	Brie	2.98
Ajax	Perrier	1.09

## ORDERS

NUM	DATE	CUST
1024	Jan 3	Zack
1025	Jan 3	Ruth
1026	Jan 4	Zack

## INCLUDES

ORD	ITEM	QUANTITY
1024	BRIE	3
1024	Perrier	6
1025	Brie	5
1026	Snails	6

INCLUDE (ORD TEM, QUANTITY)

Query: Retrieve the items in order 1024.

Select TEM

From INCLUDES

where ORD = 1024

answer

ITEM

Brie

Perrier

Another Example

ORDERS (NUM, DATE, CUST)  
INCLUDES (ORD, ITEM, QUANTITY)

Query: Retrieve the items ordered by "Zack".

```
Select ITEM  
From ORDERS, INCLUDES  
where CUST = 'Zack'  
and NUM = ORD
```

Answer

ITEM

Brie

Perrier

Snails

## SubQueries

ORDERS (NUM, DATE, CUST)

INCLUDES (ORD, ITEM, QUANTITY)

"Retrieve the items ordered by Jack."

New approach:

- First, retrieve orders placed by Jack.
- Then, retrieve the items included in these orders.

Select Item

From Includes

where Ord in (Select Num From Orders  
where Cust = 'Jack').

subquery

Ord must be in the answer to the subquery.

## Quantified Subqueries

5-

SUPPLIERS (NAME, ITEM, PRICE)

"Retrieve the supplier who charges the least for Brie."

Select Name, Price From Suppliers

where Item = 'brie'

and price  $\leq$  all (select price  
From suppliers  
where item = 'brie')

subquery

price must be  $\leq$  to all values returned  
by the subquery.

Note: Suppliers and Price are mentioned twice.  
Scoping rules resolve ambiguity.

## Another Example

6

For each item, retrieve the supplier who charges the least."

### SUPPLIERS

NAME	ITEM	PRICE
Acme	Brie	2.49
Acme	Perrier	1.19
Acme	Snails	.25
Ajax	Brie	2.98
Ajax	Perrier	1.09

### query answer

NAME	ITEM	PRICE
Acme	Brie	2.49
Acme	Snails	.25
Ajax	Perrier	1.09

7

"For each item, retrieve the supplier who charges the least."

Select Name, Item, Price

From Suppliers S1

where Price  $\leq$  all (select Price

From Suppliers S2

where S2.Item = S1.Item

subquery

For each item, the subquery returns different values.

when S1.Item = brie, the subquery returns the prices of brie.

when S1.Item = coffee, the subquery returns the prices of coffee.



## Aggregate Operators

AVG, COUNT, SUM, MIN, MAX

g: "retrieve the number of suppliers"

Select Count(NAME)

From Suppliers.

(not quite correct)

---

"retrieve the average price of Brie."

Select Avg(Price)

From Suppliers.

where item = 'Brie'

# Eliminating Duplicates

9

NAME	ITEM	PRICE
Acme	Brie	3.49
Acme	Perrier	1.19
Ajax	Perrier	1.09
Ajax	Brie	3.98

Select Name  
From Suppliers

NAME  
Acme  
Acme  
Ajax  
Ajax

Select Distinct Name  
From Suppliers

Name  
Acme  
Ajax

Select Count (Distinct Name)  
From Suppliers

} Correct way to count.

Groupin

9/1

SUPPLIERS

NAME	ITEM	PRICE
Acme	Brie	3.49
Ajax	Perrier	1.10
Swift	Coffee	2.39
Ajax	Brie	3.67
Ajax	Coffee	2.13
Acme	Perrier	1.25
Swift	Perrier	1.07
Acme	Coffee	2.68
Swift	Brie	3.55

Note: The entries are unordered.

Problem: How to group the entries by some attribute?

Select \*

From SUPPLIERS

group by name

<u>Name</u>	<u>Item</u>	<u>Price</u>
Acme	Brie	3.49
	Pennier	1.25
	CoFFee	2.68
Ajax	Pennier	1.10
	Brie	3.67
	CoFFee	2.13
Swift	CoFFee	2.39
	Pennier	1.07
	Brie	3.55

## Grouping and Aggregation

11

"What is the average price of Brie?"

Select Avg(Price) From SUPPLIERS

where Item = 'Brie'.

---

"What is the average price of each item?"

Select Item, Avg(Price) From SUPPLIERS

group by Item.

Item	avg(Price)
Brie	3.57
Perrier	1.14
Coffee	2.40

"How many items does each supplier supply?"

Select Name, count (Item)

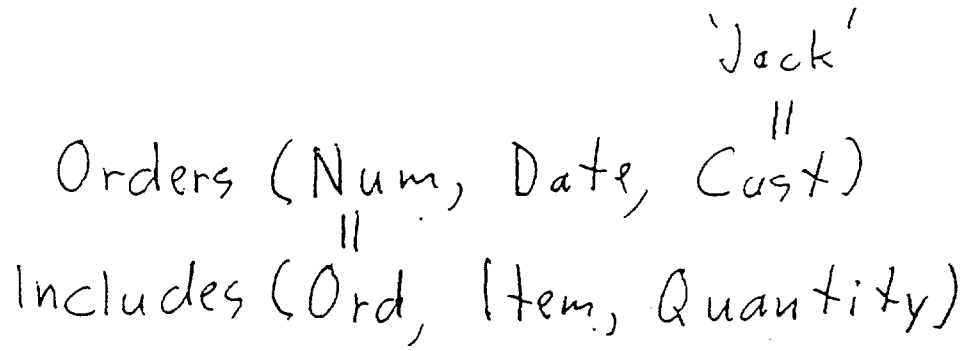
From Suppliers

group by Name

Name	count (Item)
Acme	3
Ajax	3
Swift	3

Putting it all together

"How many items are included in each of Jack's orders"?



Select count (distinct Item), Ord  
 From Orders, Includes  
 where Ord = Num and Cust = 'Jack'  
 group by Ord.

(Selection, Join, Group, Count).

SQL

Far { - Queries

text {  
- Table Definition  
- Inserting Data into Tables  
- Deleting Data  
- Updating tables

- 1000



## Data Definition

Create Table Suppliers

( Name Char(20)  
Item Char(10)  
Price Real )



Attribute Names.



Datatypes

vary from system to system.

} creating a table.

Drop Suppliers

} Destroying a table.

## Putting Data into tables

eg: Suppliers (Name, Item, Price)

Inserting single tuples Many ways to do it:

Insert into Suppliers

set to 'Acme', 'Brie', .39

Insert into Suppliers

set Name to 'Acme',  
Item to 'Brie',  
Price to 1.39

Insert into Suppliers (Item, Price, Name)

set to 'Brie', 1.39, 'Acme'

etc

## Inserting multiple tuples

Insert into Suppliers values

'Acme', 'Brie', 1.39,

'Ajax', 'Brie', 1.78,

'Ajax', 'Perrier', 2.63

## Inserting Data from other tables

Insert into MyTable

```
select * from Suppliers
```

```
where Item = 'Brie'
```

Q. "Insert all suppliers of brie into MyTable"

ven:

SUPPLIERS

NAME	ITEM	PRICE
Acme	Brie	3.49
Ajax	Perrier	1.10
Swift	Coffee	2.39
Acme	Perrier	1.25
Ajax	Brie	3.67
Swift	Perrier	1.07
Acme	Coffee	2.68

Create Table Acme\_Sells

(Item char(10), Price real);

Insert into Acme\_Sells

Select Item, Price From Suppliers  
where Name = 'Acme';

ves:

Acme\_Sells

Item	Price
Brie	3.49
Perrier	1.10
Coffee	2.68

# Deletions

Syntax:

```

Delete From Table
      where Condition

```

eg: "Acme no longer sells Brie"

```

Delete From Suppliers
      where Name = 'Acme'
      and Items = 'Brie'

```

} single delete

"Acme no longer sells anything"

```

Delete From Suppliers
      where Name = 'Acme'

```

} multiple delete

## Multiple Deletions can be Tricky

: "Delete all orders for Brie"

Brie  
||  
Includes (Ord, Item, Quantity)  
||  
Orders (Num, Date, Cust)

no tables are involved, but

delete refers to only one table.

e. Delete From Table where Condition

r.s. Select Attributes

From Tables where Condition.

eg: "Select all orders For Brie"

Select Num, Date, Cust  
From Orders, Includes  
where Ord = Num and Item = 'Brie'

What to Do?  
Use Subqueries!

Select Num, Date, Cust  
From Orders  
where Ord in (Select Num From Includes  
where Item = 'Brie')

"Delete all orders For Brie"

Delete From Orders

where Ord in (Select Num From Includes  
where item = 'Brie').



Update

yntax:

```

Update Table
set Attr1 to Val1,
  Attr2 to Val2,
  Attrn to Valn
where Condition

```

: "Change the price that Acme charges  
for Premier to \$1.00"

```

Update Suppliers
Set Price to 1.00
where Item = 'Premier'
and Name = 'Acme'

```

} single update

"Lower all Acme's prices by 10%"

Update Suppliers

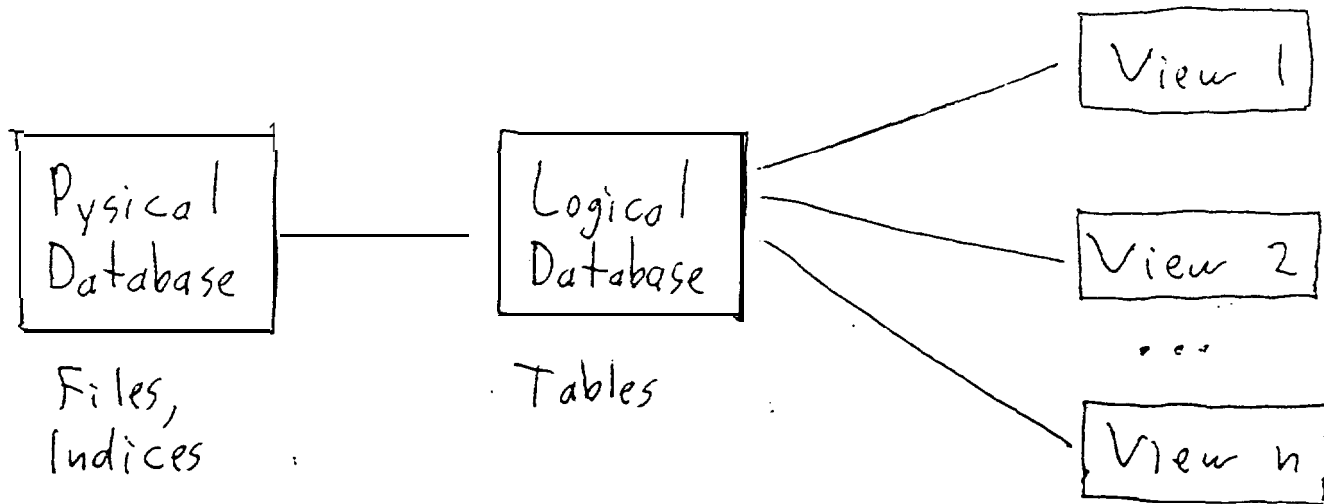
Set Price to  $.9 * \text{Price}$

where Name = 'Acme'



Multiple  
Updates

## Views (Virtual Tables)



### Advantages of Views

- Logical Data Independence
- Simplified Presentation of Data
- Different users can see the data in different ways (at the same time).
- Security / Data Hiding

- A view is "dynamic"
- A view is a "window" on the database.
- Views are updated (automatically) whenever the database is updated.
- A view is defined by a query.

Syntax:

```
Create View V  
as Query
```

Example

Given Table      Sells (Company, Item, Price)

Create View Acme\_Sells  
as Select Item, Price  
From Sells  
where Company = 'Acme' } A "window"  
on the data.

---

Create Table Acme\_Sells2 (Item Char(20,1),  
Price Float)

Insert into Acme\_Sells2  
Select Item, Price  
From Sells  
where Company = 'Acme' } A "Snapshot"  
of the data.

Sells		
company	Item	Price
Acme	Brie	2.49
Acme	Perrier	1.19
Ajax	Brie	1.09
Ajax	Perrier	2.98

Acme\_Sells

Item	Price
Brie	2.49
Perrier	1.19

Acme\_Sells2

Item	Price
Brie	2.49
Perrier	1.19

IF Sells is updated,

then **Acme\_Sells** may change,

but **Acme\_Sells2** will not.

## Evaluating Queries on Views

- Usually, a view is never materialized.
- Instead, a query to a view is "merged" with the query defining the view, to produce a new query to the underlying data base.
- Thus, it is not necessary to store a view, and a view appears to update itself automatically.

## Example

Select Price  
From Acme-Sells  
where Item = 'Perrier'

} Query  
(to view)

Create View Acme-Sells  
as Select Item, Price  
From Sells  
where Company = 'Acme'

} View  
Definition

Select Price  
From Sells  
where Item = 'Perrier'  
and Company = 'Acme'

} New Query  
(to database)



Summary

- Query Frills:
  - Subqueries, Quantifiers, Aggregation, Duplicate elimination, Grouping
- Creating Database Tables (easy).
- Updates
  - Inserts, Deletes, Modifications
  - Single tuple / Multiple tuple
  - With / Without Subqueries.
- Views
  - View Definition
  - Evaluating queries to views.