SQL SQL SQL SQL SQL SQL SQL

Eugene Wu

Fall 2018

Didn’t Lecture 3 Go Over SQL?

haha

Didn’t Lecture 3 Go Over SQL?

Two sublanguages

DDL Data Definition Language

define and modify schema (physical, logical, view)

CREATE TABLE, Integrity Constraints

DML Data Manipulation Language

get and modify data

simple SELECT, INSERT, DELETE

human-readable language

Gritty Details

DDL

NULL, Views

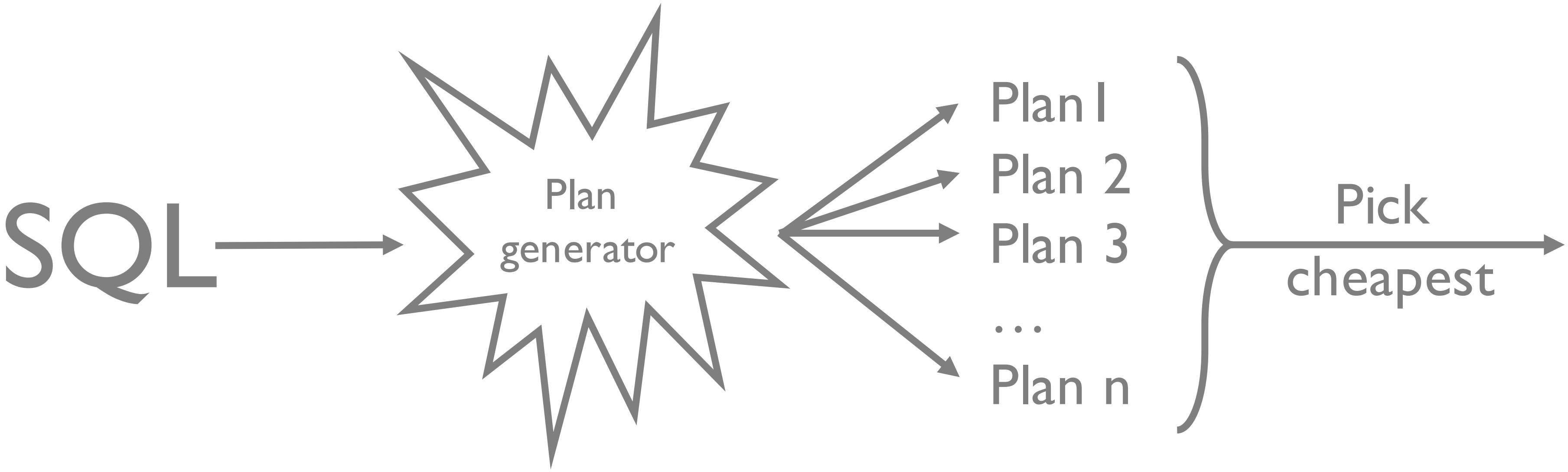
DML

Basics, SQL Clauses, Expressions, Joins, Nested Queries, Aggregation, With, Triggers

Why a declarative language (SQL)?

DBMS makes it run efficiently

- Key: precise query semantics
- Reorder/modify queries while answers stay same
- DBMS estimates costs for different evaluation plans



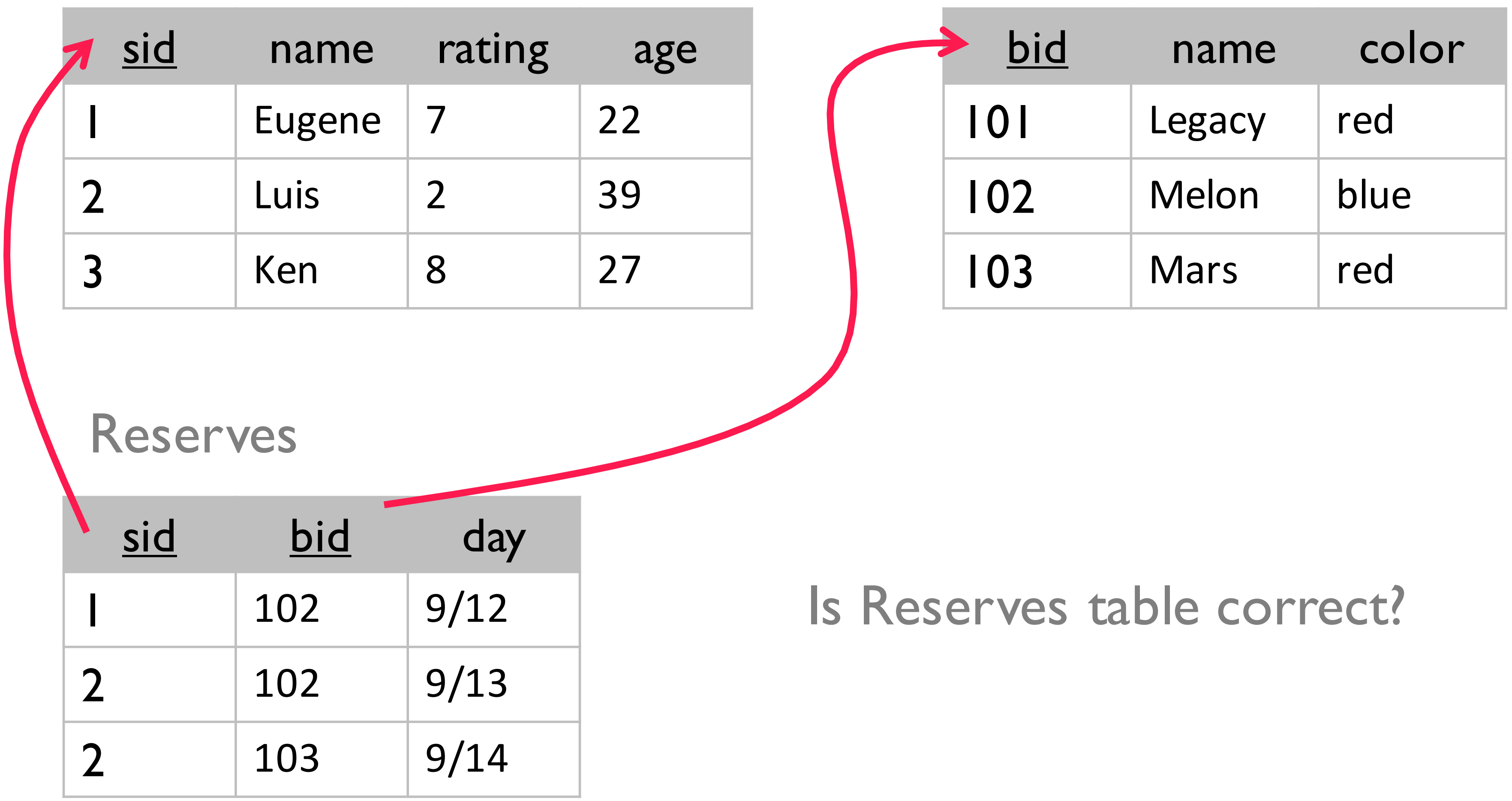
SQL Extends Relational Algebra

More expressive power than Rel Alg

- Multisets (bags) rather than sets
  - i.e. # duplicates in a table carefully accounted for
- Ordering
- NULLs
- Aggregates

Most widely used *query language*, not just relational query language

Today's Database



Sailors

sid	name	rating	age
1	Eugene	7	22
2	Luis	2	39
3	Ken	8	27

Boats

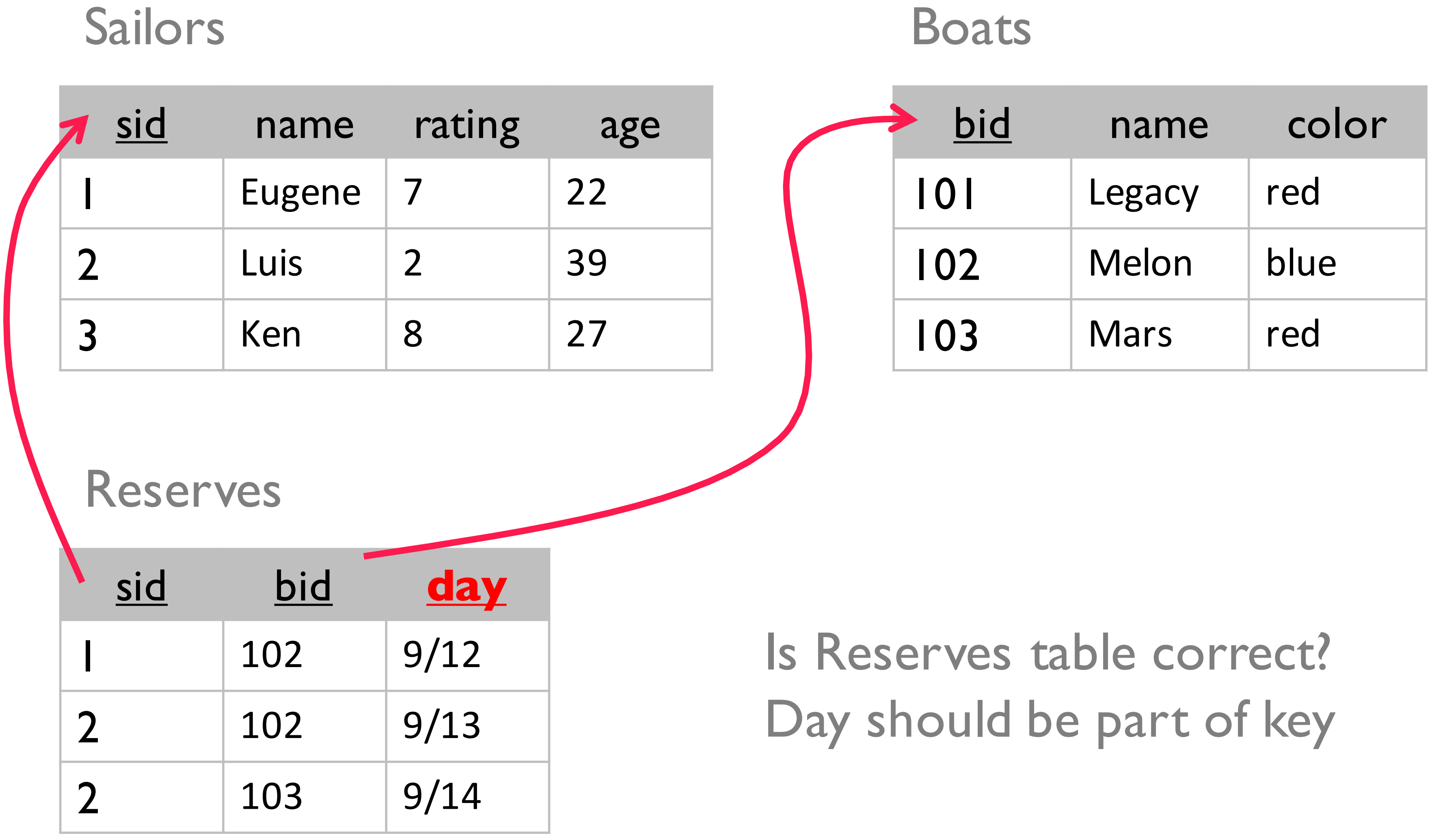
bid	name	color
101	Legacy	red
102	Melon	blue
103	Mars	red

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	103	9/14

Is Reserves table correct?

Today’s Database



Sailors

sid	name	rating	age
1	Eugene	7	22
2	Luis	2	39
3	Ken	8	27

Boats

bid	name	color
101	Legacy	red
102	Melon	blue
103	Mars	red

Reserves

sid	bid	**day**
1	102	9/12
2	102	9/13
2	103	9/14

Is Reserves table correct?

Day should be part of key

Follow along at home!

https://www.instabase.com/ewu/w4111-public/fs/Instabase%20Drive/Examples/sql.ipynb

<30 year old sailors

```
SELECT *
FROM Sailors
WHERE age < 30
```

sid	name	rating	age
1	Eugene	7	22
3	Ken	8	27

```
SELECT name, age
FROM Sailors
WHERE age < 30
```

name	age
Eugene	22
Ken	27

<30 year old sailors

```
SELECT *
FROM Sailors
WHERE age < 30
```

$\sigma_{age<30}$ (Sailors)

```
SELECT name, age
FROM Sailors
WHERE age < 30
```

$\pi_{name,\ age}$ ($\sigma_{age<30}$ (Sailors))

Multiple Relations

```
SELECT S.name
FROM   Sailors AS S, Reserves AS R
WHERE  S.sid = R.sid AND R.bid = 102
```

$$\pi_{\text{name}}(\sigma_{\text{bid}=2}(\text{Sailors} \bowtie_{\text{sid}} \text{Reserves}))$$

Sailors

sid	name	rating	age
1	**Eugene**	7	22
2	**Luis**	2	39
3	**Ken**	8	27

Reserves

sid	bid	day
1	**102**	**9/12**
2	**102**	**9/13**
2	103	9/14

Structure of a SQL Query

DISTINCT

Optional, answer should not have duplicates
Default: duplicates not removed (multiset)

target-list

List of expressions over attrs of tables in relation-list

```
SELECT    [DISTINCT] target-list
FROM      relation-list
WHERE     qualification
```

relation-list

List of relation names
Can define range-variable “AS X”

qualification

Boolean expressions

- Combined w/ AND, OR, NOT
- attr *op* const
- attr$_1$ *op* attr$_2$
- op is =, <, >, !=, etc

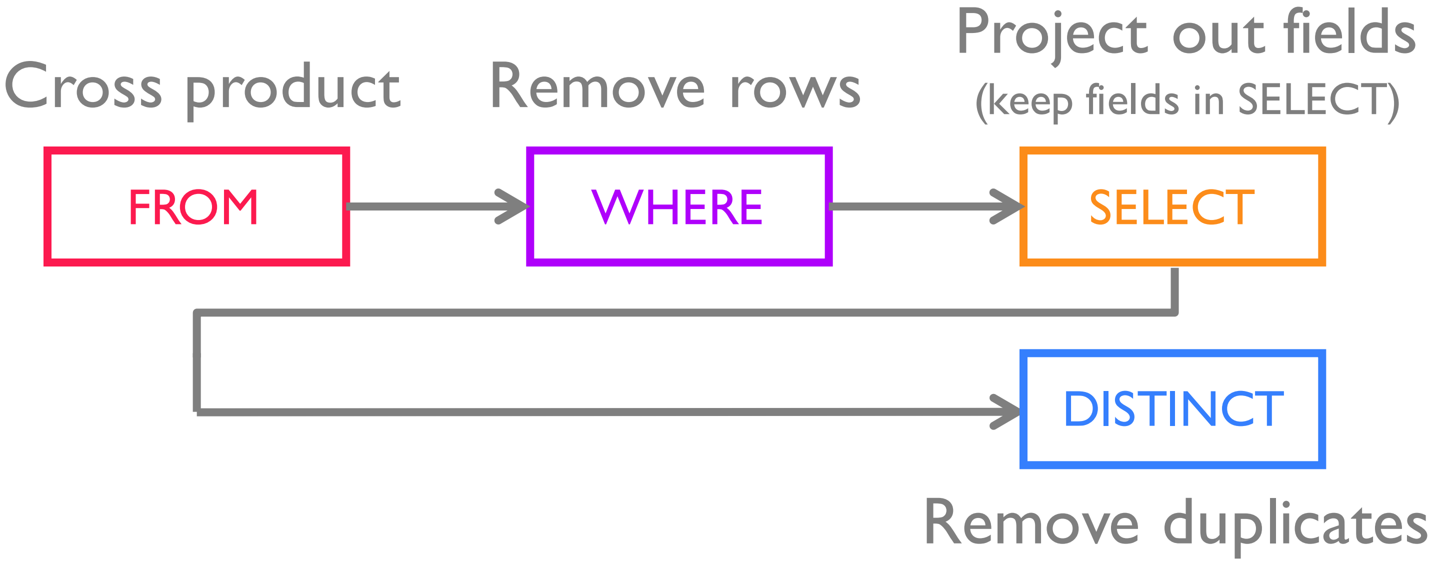
Semantics

SELECT [DISTINCT] *target-list*
FROM *relation-list*
WHERE *qualification*

FROM	compute cross product of relations
WHERE	remove tuples that fail qualifications
SELECT	remove fields not in target-list
DISTINCT	remove duplicate rows

Conceptual Query Evaluation

```
SELECT     [DISTINCT] target-list
FROM       relation-list
WHERE      qualification
GROUP BY   grouping-list
HAVING     group-qualification
```



Cross product

FROM

Remove rows

WHERE

Project out fields
(keep fields in SELECT)

SELECT

DISTINCT

Remove duplicates

Not how actually executed! Above is likely very slow

DISTINCT (vol.1)

Reserves

sid	bid	day
1	102	9/12
2	102	9/13
2	103	9/14

```
SELECT   bid
FROM     Reserves
```

bid
102
102
103

```
SELECT   DISTINCT bid
FROM     Reserves
```

bid
102
103

Sailors that reserved 1+ boats

```
SELECT  S.sid
FROM    Sailors AS S, Reserves AS R
WHERE   S.sid = R.sid
```

Would DISTINCT change anything in this query?

What if SELECT clause was SELECT S.name?

Range Variables

Disambiguate relations

- same table used multiple times (self join)

```
~~SELECT  sid~~
~~FROM    Sailors, Sailors~~
~~WHERE   age > age~~
```

```
SELECT  S1.sid
FROM    Sailors AS S1, Sailors AS S2
WHERE   S1.age > S2.age
```

Range Variables

Disambiguate relations

- same table used multiple times (self join)

```
~~SELECT  sid~~
~~FROM    Sailors, Sailors~~
~~WHERE   age > age~~
```

```
SELECT  S1.name, S1.age, S2.name, S2.age
FROM    Sailors AS S1, Sailors AS S2
WHERE   S1.age > S2.age
```