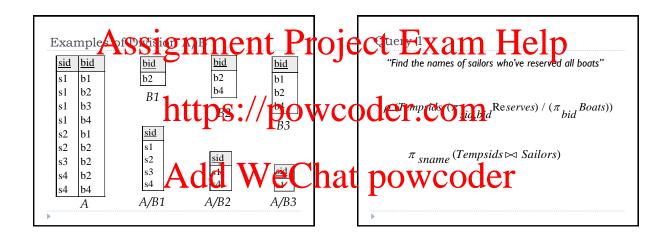


Division

- Used to answer queries such as:
 Find sailors who have reserved <u>all</u> boats.
- Let A have 2 fields, x and y; B have only field y:
- $A/B = \{ \langle x \rangle | \exists \langle x, y \rangle \in A \ \forall \langle y \rangle \in B \}$
- A/B contains all x tuples (sailors) such that for every y tuple (boat) in B, there is an xy tuple in A
- Or, if the set of y values (boats) associated with an x value (sailor) in A contains all y values in B, the x value is in A/B.
- ▶ In general, x and y can be any sets of fields (not singletons)

•



Query 2 "Find sailors who've reserved all red boats" $\rho(Temp,(\pi_{sid,bid}^{Reserves)/(\pi_{bid}^{G}(\sigma_{color='red}^{Boats})))}$ $\pi_{sname}(Temp\bowtie Sailors)$

Expressing A/B Using Basic Operators

- ▶ For A/B, compute all x values that are not disqualified by some y value in B
 - x value is disqualified if by attaching y value from B, we obtain an xy tuple that is not in A

Disqualified
$$x$$
 values: $\pi_{\chi}((\pi_{\chi}(A) \times B) - A)$

A/B: $\pi_{\chi}(A)$ - all disqualified tuples
$$\pi_{\chi}(A) - \pi_{\chi}((\pi_{\chi}(A) \times B) - A)$$

•

Division in SQL

- Not supported as primitive operator
- Need to use nested queries to express division
 - > One of the most subtle queries in SQL
 - ▶ Need to pay close attention to writing SQL division queries!
- There are two ways of writing division queries
 - Using the set EXCEPT operator (2-level nesting)
 - ▶ Without the **EXCEPT** operator (3-level nesting)

•

```
Division: Solution 1

"Find sailors who've reserved all boats."

With EXCEPT:
SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS

(
(SELECT B.bid FROM Boats B)
EXCEPT
(SELECT R.bid FROM Reserves R
WHERE R.sid=S.sid)
)
```

```
Division: Assisignment Pr
                                                                   Tind saily solo
"Find sailors who've reserved all boats.
                                                                     "Find sailors who've reserved all red boats
                                                                     With EXCEPT:
 Without EXCEPT:
                                                                     SELECT S.sname
SELECT S.sname
FROM Sailors S Sailors S su
WHERE NOT EXISTS (SELECT B.bid
                 FROM Boats B
                                                                             (SELECT B.bid FROM Boats B
                 WHERE NOT EXISTS (SELECT *
                                                                                 WHERE B.color = 'red')
 without a Reserves tuple showing S reserved B
                                   FROM Reserves R
                                                                                             Reserves R
                                                                              WHERE R.sid=S.sid)
```

```
"Find sailors who've reserved all red boats."

"Find sailors who've reserved all red boats."

Without EXCEPT:

SELECT S.sname
FROM Sailors S
WHERE NOT EXISTS (SELECT B.bid
FROM Boats B
WHERE B.color='red' AND
NOT EXISTS (SELECT *
FROM Reserves R
WHERE R.bid=B.bid
AND R.sid=S.sid))
```

```
Another Example

Movies (movie id, title, year, studio)
Actors (actor id, name, nationality)
StarsIn (actor id, movie id, character)

"Find names of actors who star in ALL movies produced by Universal in year 1990."

SELECT A.name FROM Actors A
WHERE NOT EXISTS(

SELECT M.movie_id FROM Movies M
WHERE M.year=1990 AND M.studio='Universal'

EXCEPT

SELECT S.movie_id FROM Stars_In S
WHERE S.actor_id=A.actor_id
)
```