JOINS

FLASHBACK: WE SPOKE ABOUT A TABLE CONTAINING SALES DATA

WE HAVE USED IT A BUNCH OF TIMES SO FAR

LET'S SAY WE HAVE A TABLE WITH SALES DATA

COLUMNS ARE NAMED 'STORELOCATION', 'PRODUCT', 'DATE','REVENUE'

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 18,2016	1,621.58

THIS IS A TABLE NAMED 'SALES_DATA'

THIS IS A TABLE NAMED 'SALES_DATA'

WHAT WOULD THE SQL CREATE TABLE STATEMENT FOR A TABLE LIKE THIS LOOK LIKE?..

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33

THIS IS A TABLE NAMED 'SALES_DATA'

WHAT WOULD THE SQL CREATE TABLE STATEMENT FOR A TABLE LIKE THIS LOOK LIKE?..

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33

```
CREATE TABLE Sales_Data
(
StoreLocation VARCHAR(30) NOT NULL,
Product VARCHAR(30) NOT NULL,
Date DATE NOT NULL,
Revenue DEC(10,2) NOT NULL DEFAULT 0.0,
PRIMARY KEY (StoreLocation,Product, Date)
)
```

FLASHBACK: WE SPOKE ABOUT A TABLE CONTAINING SALES DATA

BUT IN REALITY, THIS TABLE WOULD BE SPLIT INTO THREE

BUT IN REALITY, THIS TABLE WOULD BE SPLIT INTO THREE

StoreLocation	Product	Date	Revenue
Bellandur	Bananas	January 18,2016	8,236.33
Bellandur	Nutella	January 18,2016	7,455.67
Bellandur	Peanut Butter	January 18,2016	5,316.89
Bellandur	Milk	January 18,2016	2,433.76
Koramangala	Bananas	January 18,2016	9,456.01
Koramangala	Nutella	January 18,2016	3,644.33
Koramangala	Peanut Butter	January 18,2016	8,988.64
Koramangala	Milk	January 18,2016	1,621.58
Bellandur	Bananas	January 17,2016	2342.33
Bellandur	Nutella	January 17,2016	6345.10
Bellandur	Peanut Butter	January 17,2016	5673.01
Bellandur	Milk	January 17,2016	4543.98
Koramangala	Bananas	January 17,2016	8902.65
Koramangala	Nutella	January 17,2016	9114.67
Koramangala	Peanut Butter	January 17,2016	5102.05
Koramangala	Milk	January 17,2016	1299.45

'SALES_DATA'

BUT IN REALITY, THIS TABLE WOULD BE SPLIT INTO THREE

'STORES'

StoreID	StoreLocation	City
1	Bellandur	Bangalore
2	Koramangala	Bangalore
3	Deccan Gymkhana	Pune
4	Bandra	Mumbai
5	Hussain Sagar	Hyderabad
6	Powai	Mumbai
7	Koregaon Park	Pune

'PRODUCTS'

ProductID	ProductName
1	Bananas
2	Milk
3	Nutella
4	Peanut Butter
5	Marmalade
6	Oranges
7	Condensed Milk

'SALES_DATA'

StoreID	ProductID	Date	Revenue
1	1	January 18,2016	8,236.33
1	3	January 18,2016	7,455.67
1	4	January 18,2016	5,316.89
1	2	January 18,2016	2,433.76
2	1	January 18,2016	9,456.01

FLASHBACK: WE HAVE COME ACROSS EXAMPLES WHERE WE CONNECT TWO TABLES VIA A MATCH COLUMN

EXAMPLE

FIND STUDENTID, FIRST, LAST NAMES FOR STUDENTS WHOSE DORMITORY IS 'AKBAR HALL'

FIND STUDENTID, FIRST, LAST NAMES FOR STUDENTS WHOSE DORMITORY IS 'AKBAR HALL'

THIS IS A TABLE NAMED 'CAMPUS_HOUSING'

StudentID	DormitoryName	AptNumber
1	Gandhi House	110
2	Akbar Hall	231
3	Gandhi House	345
4	NULL	NULL

COLUMNS ARE NAMED 'STUDENTID', 'DORMITORYNAME', 'APTNUMBER'

THIS IS A TABLE NAMED 'STUDENTS'

StudentID	FirstName	LastName	Gender	Email
1	Janani	Ravi	F	janani@loonycorn.com
2	Swetha	Kolalapudi	F	swetha@loonycorn.com
3	Navdeep	Singh	M	navdeep@loonycorn.com
4	Vitthal	Srinivasan	M	vitthal@loonycorn.com

COLUMNS ARE NAMED 'STUDENTID', 'FIRSTNAME', 'LASTNAME', 'GENDER' AND 'EMAIL'

FIND STUDENTID, FIRST, LAST NAMES FOR STUDENTS WHOSE DORMITORY IS 'AKBAR HALL'

SELECT	WHICH COLUMNS?	STUDENTID, FIRSTNAME, LASTNAME
FROM	WHICH TABLES?	STUDENTS, CAMPUS_HOUSING
WHERE	WHICH ROWS?	WHERE DORM IS CALLED 'AKBAR HALL'

FIND STUDENTID, FIRST, LAST NAMES FOR STUDENTS WHOSE DORMITORY IS 'AKBAR HALL'

```
SELECT    STUDENTS.STUDENTID,FIRSTNAME,
                    LASTNAME

FROM      STUDENTS S, CAMPUS_HOUSING C

WHERE     DORMITORYNAME = 'AKBAR HALL'

      AND S.STUDENTID = C.STUDENTID;
```

FIND STUDENTID, FIRST, LAST NAMES FOR STUDENTS WHOSE DORMITORY IS 'AKBAR HALL'

StudentID	FirstName	LastName
2	Swetha	Kolalapudi

FLASHBACK: WE HAVE COME ACROSS EXAMPLES WHERE WE CONNECT TWO TABLES VIA A MATCH COLUMN

SUCH CONNECTS HAVE RICH SUPPORT IN SQL - VIA

JOINS

JOINS

ARE INCREDIBLY USEFUL WAYS TO CONNECT
TABLES AND QUERY THEM AS A UNIT

JOINS

ARE INCREDIBLY USEFUL WAYS TO CONNECT
TABLES AND QUERY THEM AS A UNIT

THE EXAMPLE WE JUST COVERED IS SIMPLY ONE TYPE OF JOIN

JOINS

ARE INCREDIBLY USEFUL WAYS TO CONNECT
TABLES AND QUERY THEM AS A UNIT

THE EXAMPLE WE JUST COVERED IS
SIMPLY ONE TYPE OF JOIN

SQL HAS JOIN KEYWORDS TO SUPPORT
SUCH QUERIES OUT-OF-THE-BOX

JOINS

CROSS JOIN
"CARTESIAN JOIN"

INNER JOIN

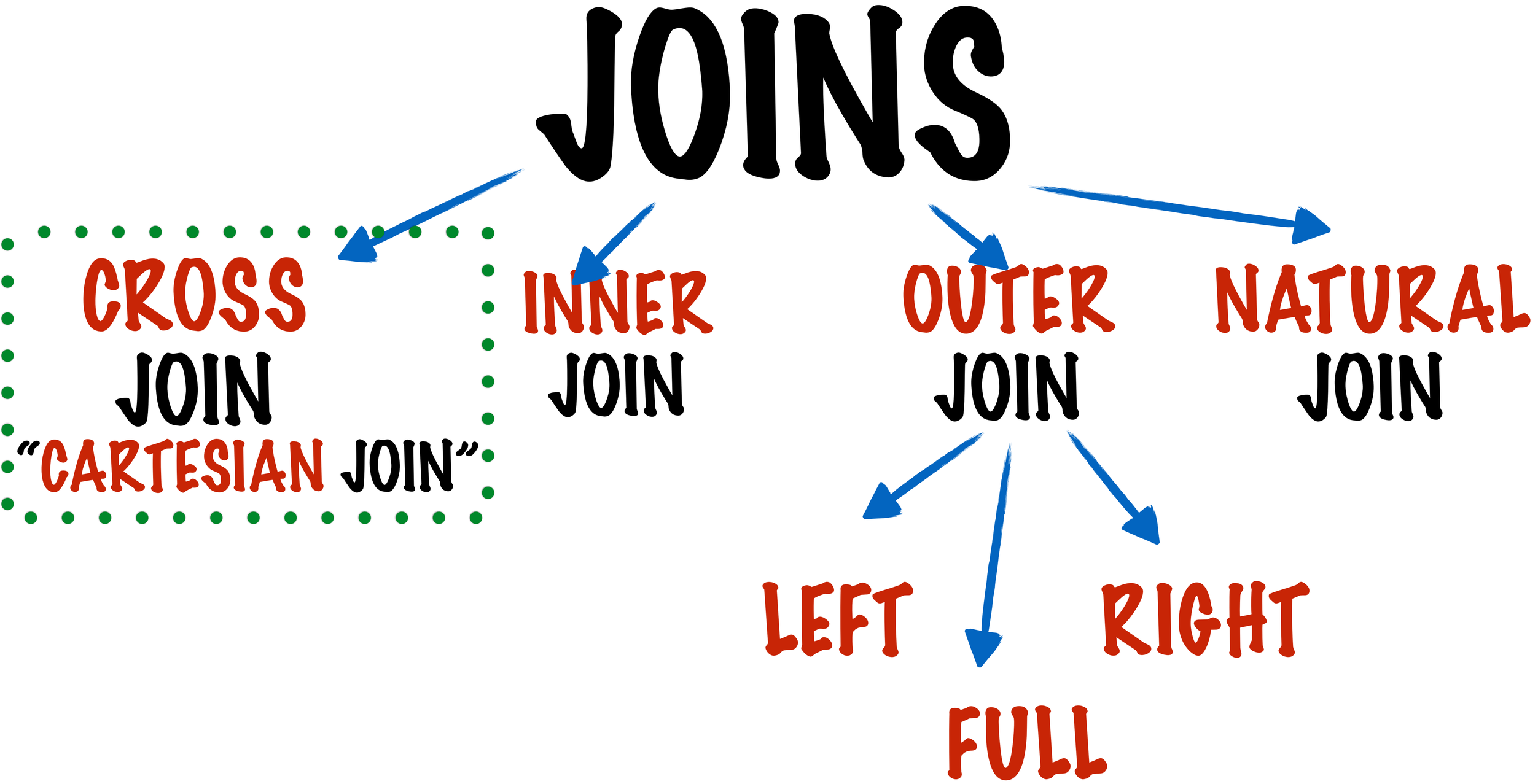
OUTER JOIN

NATURAL JOIN

ARE INCREDIBLY USEFUL WAYS TO CONNECT TABLES AND QUERY THEM AS A UNIT

THE EXAMPLE WE JUST COVERED IS SIMPLY ONE TYPE OF JOIN

SQL HAS JOIN KEYWORDS TO SUPPORT SUCH QUERIES OUT-OF-THE-BOX


JOINS
CROSS
JOIN
"CARTESIAN JOIN"
INNER
JOIN
OUTER
JOIN
NATURAL
JOIN
LEFT
FULL
RIGHT