SQL

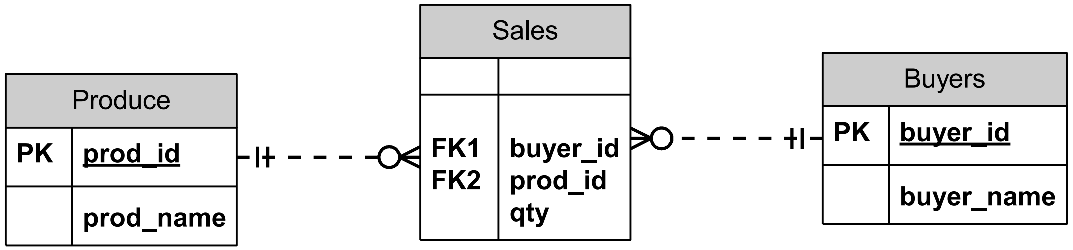
- **Structured Query Language**

Joining tables

- Aliases for table names
- Introduction to Joins
- Using Inner Joins
- Using Outer Joins
- Using Cross Joins
- Joining more than two tables
- Joining a table to itself

Joindb database

Joindb Database Diagram



Produce	
PK	<u>prod_id</u>
	prod_name

Sales	
FK1 FK2	buyer_id prod_id qty

Buyers	
PK	<u>buyer_id</u>
	buyer_name

Using aliases for table names

```
USE joindb
 SELECT buyer_name, sales.buyer_id, qty
 FROM buyers, sales
 WHERE buyers.buyer_id = sales.buyer_id
```

```
USE joindb
SELECT buyer_name, s.buyer_id, qty
 FROM buyers AS b, sales AS s
 WHERE b.buyer_id = s.buyer_id
```

Using aliases for table names

```
USE joindb
 SELECT buyer_name, sales.buyer_id, qty
 FROM buyers  INNER JOIN sales
  ON buyers.buyer_id = sales.buyer_id
```

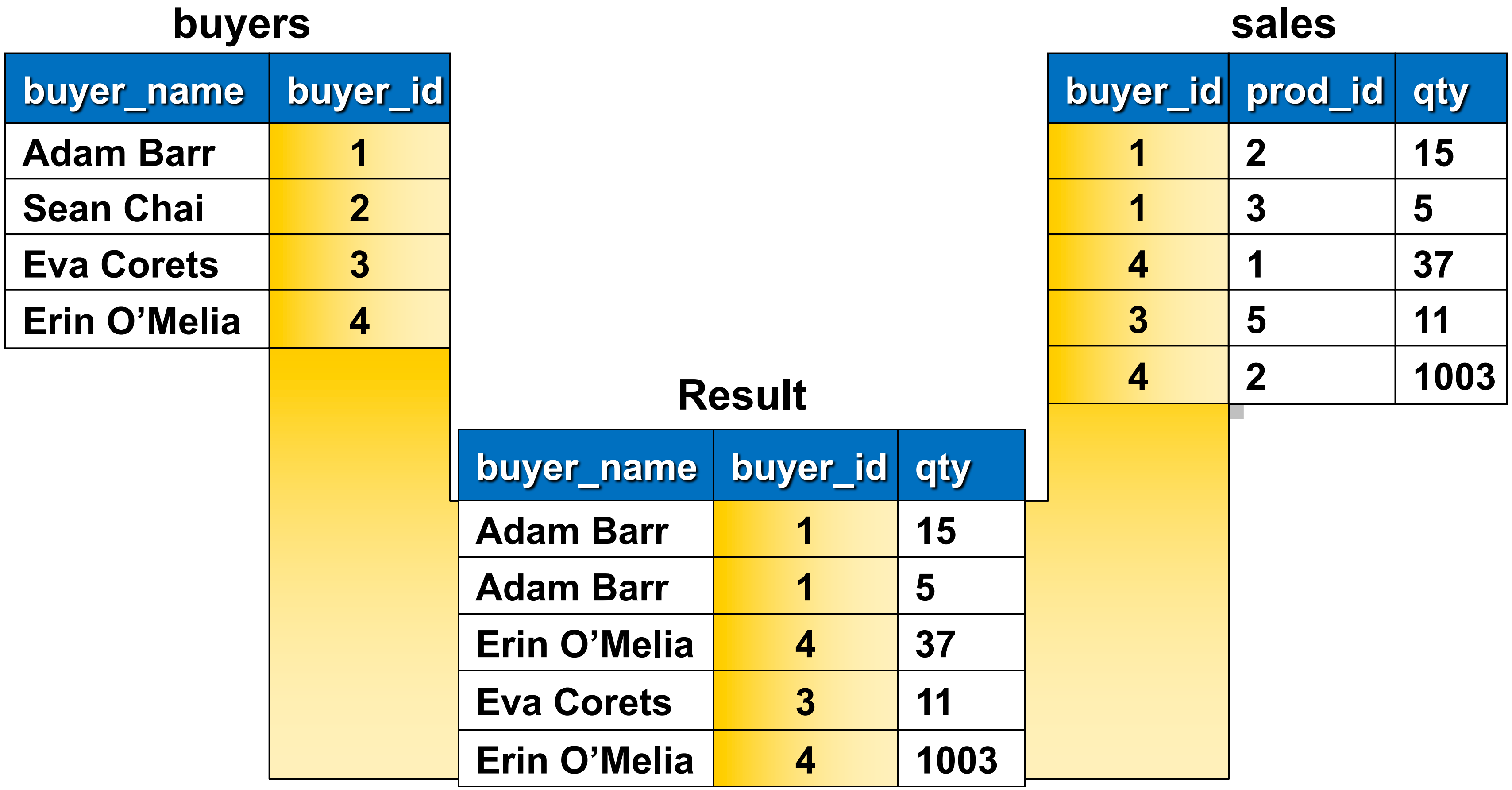
```
USE joindb
SELECT buyer_name, s.buyer_id, qty
 FROM buyers AS b  INNER JOIN sales AS s
  ON b.buyer_id = s.buyer_id
```

Introduction to Joins

- **Selects specific columns from multiple tables**
  - JOIN keyword specifies that tables are joined and how to join them
  - ON keyword specifies join condition
- **Queries two or more tables to produce a result set**
  - Use primary and foreign keys as join conditions
  - Use columns common to specified tables to join tables

INNER JOIN

```
SELECT buyer_name, sales.buyer_id, qty
FROM buyers  INNER JOIN sales
ON buyers.buyer_id = sales.buyer_id
```



buyers

buyer_name	buyer_id
Adam Barr	1
Sean Chai	2
Eva Corets	3
Erin O’Melia	4

sales

buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

Result

buyer_name	buyer_id	qty
Adam Barr	1	15
Adam Barr	1	5
Erin O’Melia	4	37
Eva Corets	3	11
Erin O’Melia	4	1003

Example

- **Select the names of products and the companies that supply the products. Products without listed suppliers and suppliers without current products are not included in the result set.**

```
SELECT productname, companyname
 FROM products
 INNER JOIN suppliers
  ON products.supplierid = suppliers.supplierid
```

Example

- **Select the names of customers who placed orders after 1/1/98.**
  - Notice that a WHERE clause is used to restrict the rows that are returned in the result set.

```
SELECT DISTINCT companyname
 FROM orders
 INNER JOIN customers
  ON orders.customerid = customers.customerid
 WHERE orderdate > '1/1/98'
```

OUTER JOIN

```
SELECT buyer_name, sales.buyer_id, qty
 FROM buyers  LEFT OUTER JOIN sales
  ON buyers.buyer_id = sales.buyer_id
```

buyers

buyer_name	buyer_id
Adam Barr	1
Sean Chai	2
Eva Corets	3
Erin O’Melia	4

sales

buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

Result

buyer_name	buyer_id	qty
Adam Barr	1	15
Adam Barr	1	5
Erin O’Melia	4	37
Eva Corets	3	11
Erin O’Melia	4	1003
Sean Chai	NULL	NULL

Example

- **Select all customers with order dates**
  - NULL in the orderdate column is returned in the result set for customers who have not placed an order

```
SELECT companyname, customers.customerid, orderdate
 FROM customers
 LEFT OUTER JOIN orders
  ON customers.customerid = orders.customerid
```

CROSS JOIN

```
SELECT buyer_name, qty
 FROM buyers
 CROSS JOIN sales
```

buyers

buyer_id	buyer_name
1	Adam Barr
2	Sean Chai
3	Eva Corets
4	Erin O’Melia

sales

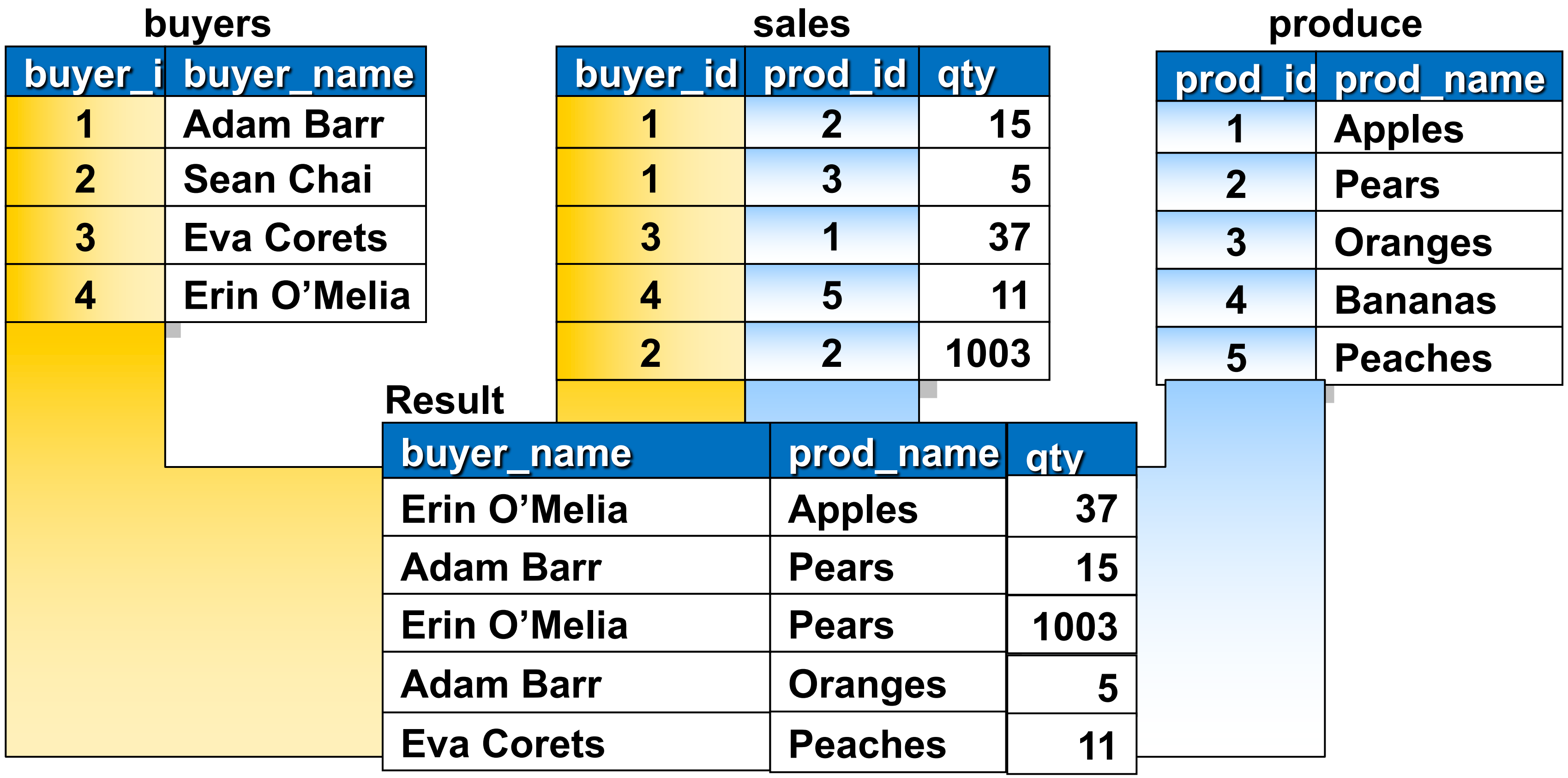
buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

Result

buyer_name	qty
Adam Barr	15
Adam Barr	5
Adam Barr	37
Adam Barr	11
Adam Barr	1003
Sean Chai	15
Sean Chai	5
Sean Chai	37
Sean Chai	11
Sean Chai	1003
Eva Corets	15
...	...

Joining more than two tables

```
SELECT buyer_name, prod_name, qty
 FROM buyers
 INNER JOIN sales
  ON buyers.buyer_id = sales.buyer_id
 INNER JOIN produce
  ON sales.prod_id = produce.prod_id
```



buyers

buyer_i	buyer_name
1	Adam Barr
2	Sean Chai
3	Eva Corets
4	Erin O’Melia

sales

buyer_id	prod_id	qty
1	2	15
1	3	5
3	1	37
4	5	11
2	2	1003

produce

prod_id	prod_name
1	Apples
2	Pears
3	Oranges
4	Bananas
5	Peaches

Result

buyer_name	prod_name	qty
Erin O’Melia	Apples	37
Adam Barr	Pears	15
Erin O’Melia	Pears	1003
Adam Barr	Oranges	5
Eva Corets	Peaches	11

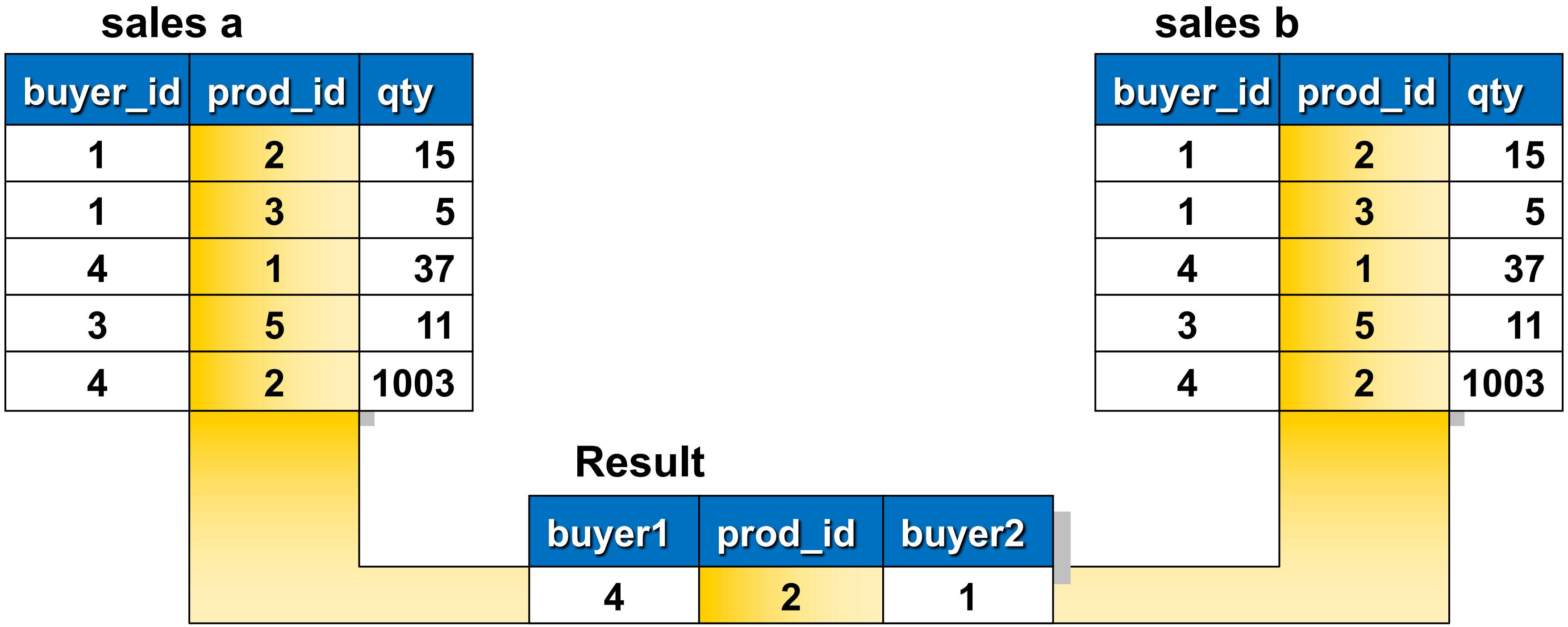
Example

- **Select products ordered in 1996-07-08.**

```
SELECT orderdate, productname
 FROM orders AS O
 INNER JOIN [order details] AS OD
  ON O.orderid = OD.orderid
 INNER JOIN products AS P
  ON OD.productid = P.productid
 WHERE orderdate = '7/8/96'
```

Joining a table to itself

```
SELECT a.buyer_id AS buyer1, a.prod_id
      ,b.buyer_id AS buyer2
 FROM sales AS a
 JOIN sales AS b
  ON a.prod_id = b.prod_id
WHERE  a.buyer_id > b.buyer_id
```



sales a

buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

sales b

buyer_id	prod_id	qty
1	2	15
1	3	5
4	1	37
3	5	11
4	2	1003

Result

buyer1	prod_id	buyer2
4	2	1

Example

- **Select pairs of employees who have the same job title**

```
SELECT a.employeeid, a.lastname,10 AS name
       ,a.title AS title
       ,b.employeeid, b.lastname AS name
       ,b.title AS title
 FROM employees AS a
 INNER JOIN employees AS b
  ON a.title = b.title
 WHERE a.employeeid < b.employeeid
```

UNION

- **Use the UNION operator to create a single result set from multiple Queries**
- **Each Query Must Have:**
  - Similar data types
  - Same number of columns
  - Same column order in select list

```
SELECT  (firstname + ' ' + lastname) AS name
       ,city, postalcode
 FROM employees
UNION
SELECT companyname, city, postalcode
 FROM customers
```