Column names and JDBC

Ambiguous names?

Remember this?

```
sqlite> .schema
CREATE TABLE productGroup(
   id INT PRIMARY KEY NOT NULL, name text
);

CREATE TABLE product(
   nr INT primary key not null, name text, price REAL,
   alcohol REAL, volume INT, productGroupId INT, type text,
   foreign key(productGroupId) references productGroup(id)
);
```

name in both the productGroup and product tables

```
--joining product and product group might end up with two
columns called “name”
--We must qualify the name column, or we’d get this:
sqlite> SELECT name, price, name FROM
>          product JOIN  productgroup
>        ON product.productgroupid = productgroup.id LIMIT 3;
Error: ambiguous column name: name
```

name in both the productGroup and product tables

```
-- using full names (including table names):
sqlite> SELECT product.name, price, productgroup.name
   ...> FROM product JOIN productgroup
   ...> ON product.productgroupid = productgroup.id LIMIT 3;
name|price|name
Johanneshof Reinisch Pinot Noir|143.0|Rött vin
Dobogó Tokaji Furmint|187.0|Vitt vin
Château de Cazeneuve Carignan|250.0|Rött vin

-- But, now we end up with two columns named “name”!
```

name in both the productGroup and product tables

```
// Imagine a ResultSet with the following three rows:
(name|price|name)
Johanneshof Reinisch Pinot Noir|143.0|Rött vin
Dobogó Tokaji Furmint|187.0|Vitt vin
Château de Cazeneuve Carignan|250.0|Rött vin

// What would rs.getString("name") return? The product’s
// name or the product group’s name?
```

name in both the productGroup and product tables

```
// Solution 1, use the overloaded version of getString(int):
SELECT product.name, price, productGroup.name....
           col 1      col2          col3

Johanneshof Reinisch Pinot Noir|143.0|Rött vin
Dobogó Tokaji Furmint|187.0|Vitt vin
Château de Cazeneuve Carignan|250.0|Rött vin

// Product's name:          rs.getString(1)
// Product's price:         rs.getString(2)
// Product's product group: rs.getString(3)
```

name in both the productGroup and product tables

```
// Solution 2, use aliases in the SQL SELECT:
SELECT product.name, price, productGroup.name AS pr_group...
           name      price          pr_group

Johanneshof Reinisch Pinot Noir|143.0|Rött vin
Dobogó Tokaji Furmint|187.0|Vitt vin
Château de Cazeneuve Carignan|250.0|Rött vin

// Product's name:          rs.getString("name")
// Product's price:         rs.getString("price")
// Product's product group: rs.getString("pr_group")
```

A SELECT kind of creates a new table

You can think of a SELECT as if it creates a new table from which it does SELECT * FROM <newtable>

The SELECT names the columns to include in the new table (and a JOIN might name from where the data comes - could be other tables). This is sometimes called “projection”.

```
SELECT product.name, productgroup.name FROM product JOIN
  productgroup ON product_id = productgroup.id LIMIT 1;
name                              name
--------------------------------  ----------
Johanneshof Reinisch Pinot Noir   Rött vin
```

Further reading

- https://en.wikipedia.org/wiki/Alias_(SQL)
- https://www.sqlite.org/lang_select.html
- https://www.sqlite.org/syntax/result-column.html
- https://tinyurl.com/y9m6ado3 (google books)
- https://www.cs.mun.ca/java-api-1.5/guide/jdbc/getstart/resultset.html