Storing query commands

Using VIEWs

Preamble

- To know the context of this presentation, look at
  - ‘Introduction to sample case study – Joe’s Yard’

VIEWs

- Store a compound statement as a new virtual set of rows.
- These can be used for
  - Ensuring privacy
  - Limiting access (security)
  - Making further compound queries simpler.

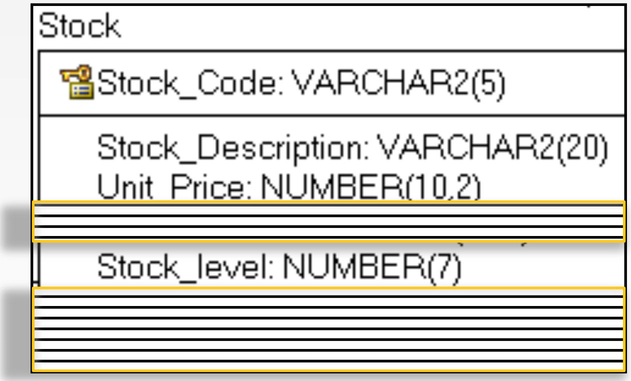
Customising views

- Sometimes I want to show aspects of my data to other users and even allow them to update them.
- I can create a VIEW of the data for manipulation by other users.
- E.g. I want the customer to be able to see certain information from the stock table, but not all of it.
- I can create a view and let customers SELECT from it.

Views

- CREATE VIEW CUSTSTOCK AS
  (select stock_code, stock_description, unit_price, stock_level from stock);

- Grant select on custstock to public;
  - Now everyone can query these fields.

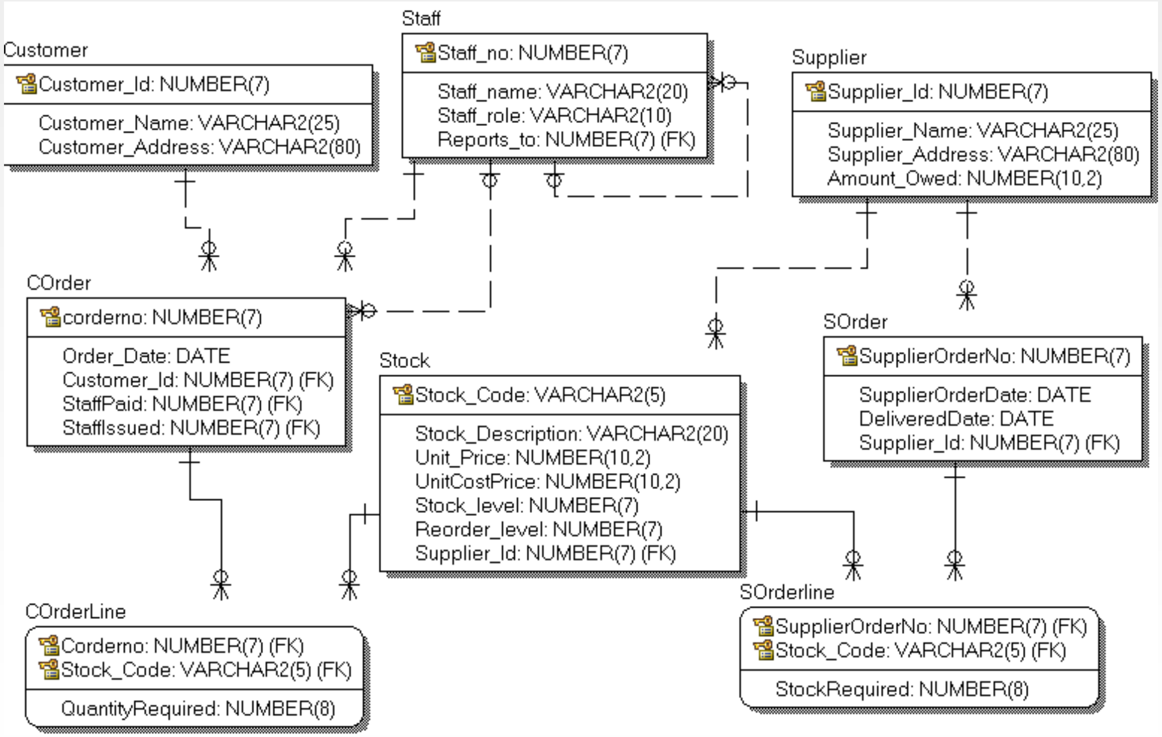


Stock

Stock_Code: VARCHAR2(5)

Stock_Description: VARCHAR2(20)
Unit_Price: NUMBER(10,2)
[redacted]
Stock_level: NUMBER(7)
[redacted]

The tables…



Customer
Customer_Id: NUMBER(7)
Customer_Name: VARCHAR2(25)
Customer_Address: VARCHAR2(80)

Staff
Staff_no: NUMBER(7)
Staff_name: VARCHAR2(20)
Staff_role: VARCHAR2(10)
Reports_to: NUMBER(7) (FK)

Supplier
Supplier_Id: NUMBER(7)
Supplier_Name: VARCHAR2(25)
Supplier_Address: VARCHAR2(80)
Amount_Owed: NUMBER(10,2)

COrder
corderno: NUMBER(7)
Order_Date: DATE
Customer_Id: NUMBER(7) (FK)
StaffPaid: NUMBER(7) (FK)
StaffIssued: NUMBER(7) (FK)

Stock
Stock_Code: VARCHAR2(5)
Stock_Description: VARCHAR2(20)
Unit_Price: NUMBER(10,2)
UnitCostPrice: NUMBER(10,2)
Stock_level: NUMBER(7)
Reorder_level: NUMBER(7)
Supplier_Id: NUMBER(7) (FK)

SOrder
SupplierOrderNo: NUMBER(7)
SupplierOrderDate: DATE
DeliveredDate: DATE
Supplier_Id: NUMBER(7) (FK)

COrderLine
Corderno: NUMBER(7) (FK)
Stock_Code: VARCHAR2(5) (FK)
QuantityRequired: NUMBER(8)

SOrderline
SupplierOrderNo: NUMBER(7) (FK)
Stock_Code: VARCHAR2(5) (FK)
StockRequired: NUMBER(8)

Possible views

- The foreman needs to see the set of stock items that needs to be issued for an outstanding order.
    - stock code, stock description, quantityrequired and corderno of any order that has been paid for, but not issued.

```
CREATE or replace VIEW waitingorders AS
SELECT stock_code,  stock_description,
  quantityrequired, corderno, staffissued
FROM corderline
JOIN stock using (stock_code)
JOIN corder using (corderno)
where staffpaid is not null and staffissued is null;
```

Possible views

- He can see the details as follows:
  - SELECT * from waitingorders;
  - Or
  - SELECT * from waitingorders where corderno = 108;
- He needs to be able to ***update*** staffissued.
  - Don’t do this using a view!!!
- Views can be DROPped or CREATE or REPLACEd.