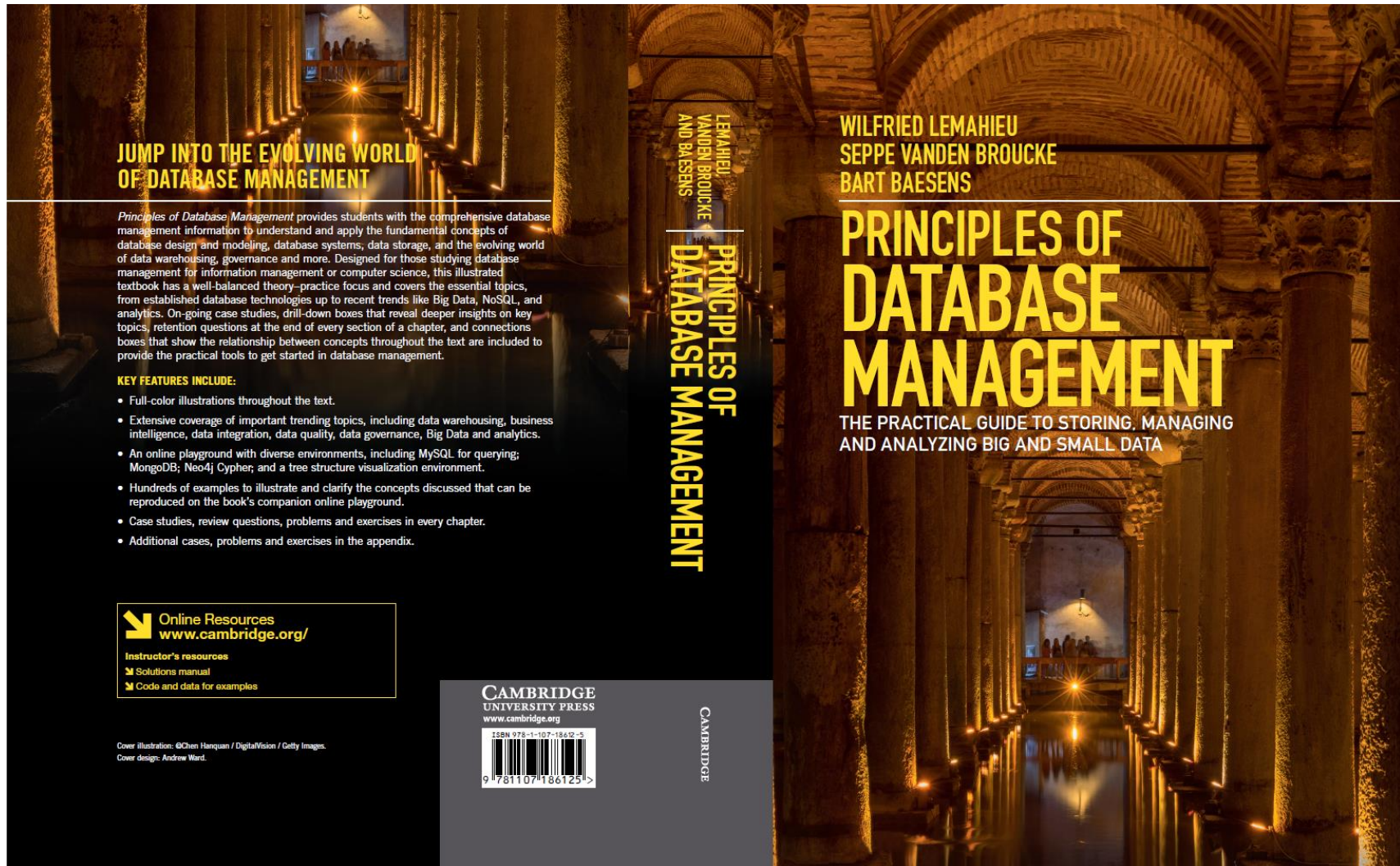


Relational Databases: Structured Query Language (SQL)



www.pdbmbook.com

Introduction

- Relational Database Management Systems and SQL
- SQL Data Definition Language
- SQL Data Manipulation Language
- SQL Views
- SQL Indexes
- SQL Privileges
- SQL for Metadata Management

Relational Database Management Systems and SQL

- Key Characteristics of SQL
- Three-Level Database Architecture

Key Characteristics of SQL

- First version, SQL-86 in 1986, most recent version in 2016 (SQL:2016)
- Accepted by the American National Standards Institute (ANSI) in 1986 and by the International Organization for Standardization (ISO) in 1987
- Each vendor provides own implementation (SQL dialect) of SQL

Key Characteristics of SQL

- Set-oriented and declarative
- Free form language
- Case insensitive
- Can be used interactively from a command prompt or executed by a program

Key Characteristics of SQL

The screenshot displays the MySQL Workbench interface. The left sidebar contains the 'Navigator' pane with sections for MANAGEMENT, INSTANCE, PERFORMANCE, and SCHEMAS. The 'SCHEMAS' section is expanded, showing the 'purchaseadmin' database with tables like 'po_line', 'product', 'purchase_order', 'supplier', and 'supplies'. The main window shows a SQL query in the 'Query 1' editor:

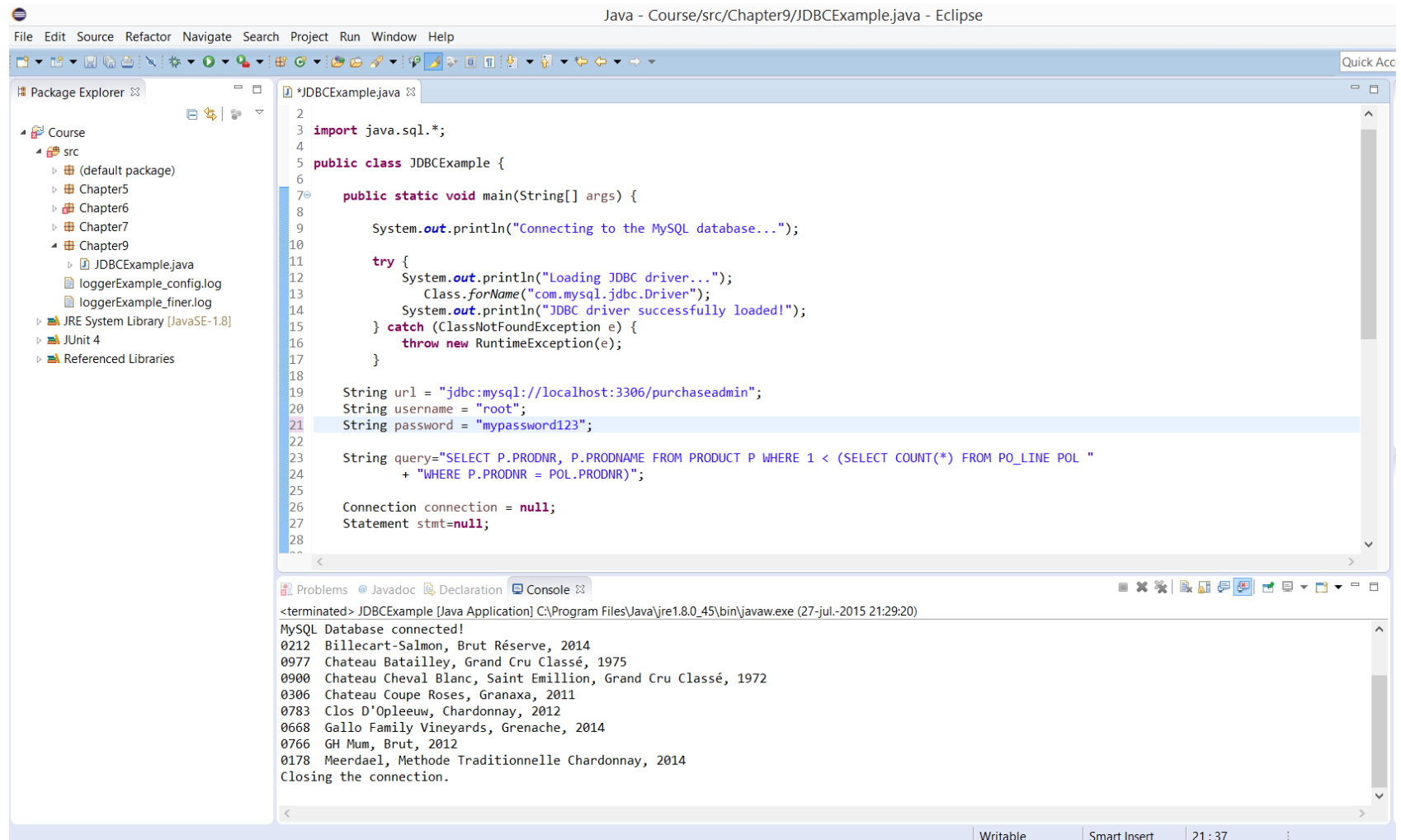
```
1 SELECT P.PRODNR, P.PRODNAME FROM PRODUCT P
2 WHERE 1 <
3     (SELECT COUNT(*)
4      FROM PO_LINE POL
5      WHERE P.PRODNR = POL.PRODNR)
6
```

Below the query editor is the 'Result Grid' showing the results of the query:

PRODNR	PRODNAME
0212	Billecart-Salmon, Brut Réserve, 2014
0977	Chateau Bataille, Grand Cru Classé, 1975
0900	Chateau Cheval Blanc, Saint Emilion, Grand Cru Classé, 1972
0306	Chateau Coupe Roses, Granaxa, 2011
0783	Clos D'Opleeuw, Chardonnay, 2012
0668	Gallo Family Vineyards, Grenache, 2014
0766	GH Mum, Brut, 2012
0178	Meerdael, Methode Traditionnelle Chardonnay, 2014

The bottom of the interface shows the 'Output' pane with a message: '1 16:47:29 SELECT P.PRODNR, P.PRODNAME FROM PRODUCT P WHERE 1 < (SELECT COUNT(*) FROM PO_LINE P... 8 row(s) returned 0.000 sec / 0.000 sec'.

Key Characteristics of SQL



The screenshot displays the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Package Explorer on the left shows a project structure with a 'src' folder containing 'JDBCExample.java'. The main editor window shows the code for 'JDBCExample.java', which imports 'java.sql.*' and defines a 'main' method. The code connects to a MySQL database using 'jdbc:mysql://localhost:3306/purchaseadmin' with username 'root' and password 'mypassword123'. It then executes a SQL query to select product names and IDs from the 'PRODUCT' table, ordered by product ID. The console at the bottom shows the output of the program, including the connection status and the results of the query.

```
2
3 import java.sql.*;
4
5 public class JDBCExample {
6
7     public static void main(String[] args) {
8
9         System.out.println("Connecting to the MySQL database...");
10
11         try {
12             System.out.println("Loading JDBC driver...");
13             Class.forName("com.mysql.jdbc.Driver");
14             System.out.println("JDBC driver successfully loaded!");
15         } catch (ClassNotFoundException e) {
16             throw new RuntimeException(e);
17         }
18
19         String url = "jdbc:mysql://localhost:3306/purchaseadmin";
20         String username = "root";
21         String password = "mypassword123";
22
23         String query="SELECT P.PRODNR, P.PRODNAME FROM PRODUCT P WHERE 1 < (SELECT COUNT(*) FROM PO_LINE POL "
24             + "WHERE P.PRODNR = POL.PRODNR)";
25
26         Connection connection = null;
27         Statement stmt=null;
28
29     }
```

Problems Javadoc Declaration Console

<terminated> JDBCExample [Java Application] C:\Program Files\Java\jre1.8.0_45\bin\javaw.exe (27-Jul-2015 21:29:20)

MySQL Database connected!

0212 Billecart-Salmon, Brut Réserve, 2014

0977 Chateau Batailley, Grand Cru Classé, 1975

0900 Chateau Cheval Blanc, Saint Emillion, Grand Cru Classé, 1972

0306 Chateau Coupe Roses, Granaxa, 2011

0783 Clos D'Opleeuw, Chardonnay, 2012

0668 Gallo Family Vineyards, Grenache, 2014

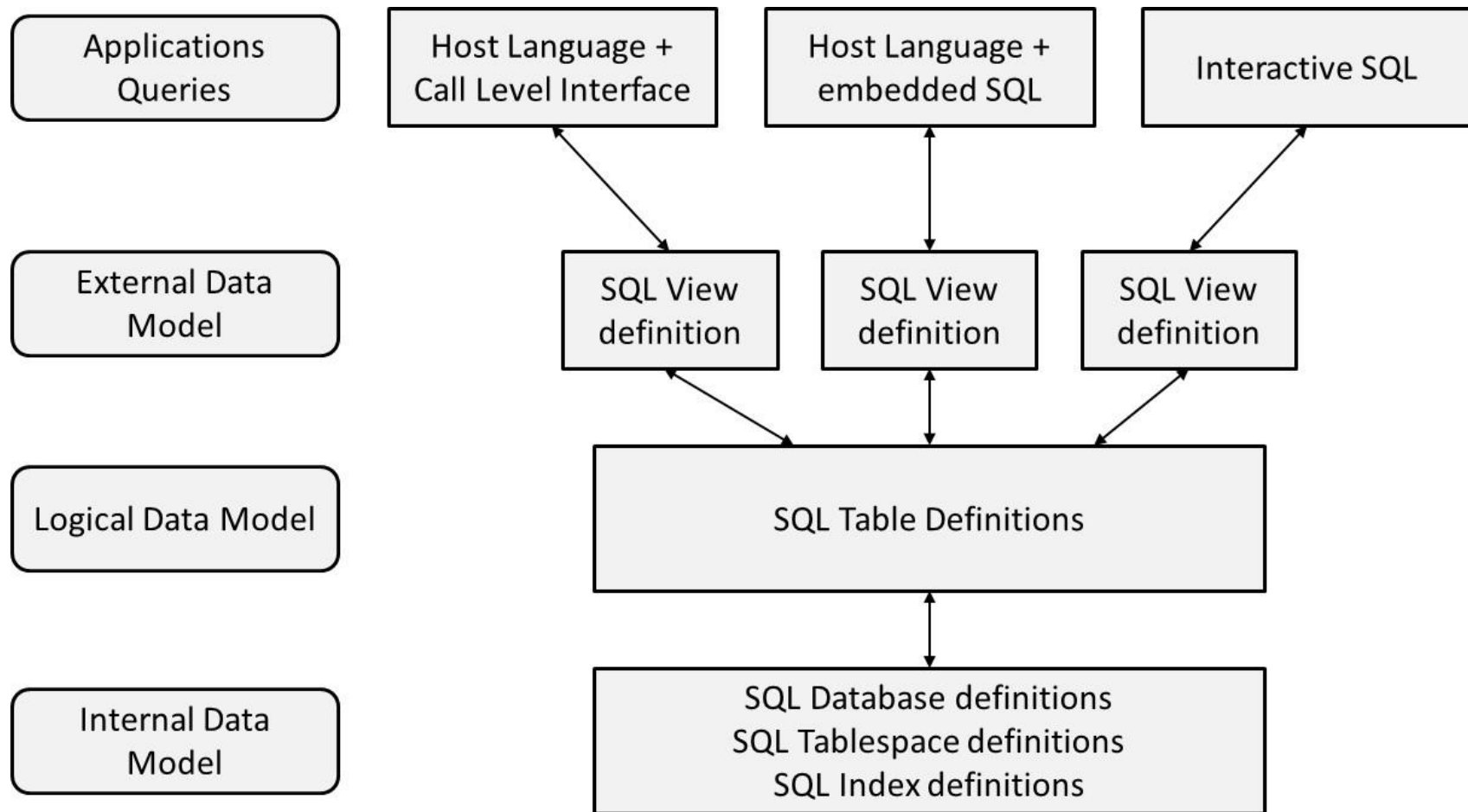
0766 GH Mum, Brut, 2012

0178 Meerdael, Methode Traditionnelle Chardonnay, 2014

Closing the connection.

Writable Smart Insert 21:37

Three-Level Database Architecture



SQL Data Definition Language

- Key DDL concepts
- DDL Example
- Referential Integrity Constraints
- DROP and ALTER command

Key DDL Concepts

- SQL Schema: grouping of tables and other database objects such as views, constraints and indexes which logically belong together

**CREATE SCHEMA PURCHASE AUTHORIZATION
BBAESENS**

- SQL table implements a relation from the relational model

**CREATE TABLE PRODUCT ...
CREATE TABLE PURCHASE.PRODUCT ...**

Key DDL Concepts

Data Type	Description
CHAR(n)	Holds a fixed length string with size n
VARCHAR(n)	Holds a variable length string with maximum size n
SMALLINT	Small integer (no decimal) between -32768 to 32767
INT	Integer (no decimal) between -2147483648 to 2147483647
FLOAT(n,d)	Small number with a floating decimal point. The total maximum number of digits is n with a maximum of d digits to the right of the decimal point.
DOUBLE(n,d)	Large number with a floating decimal point. The total maximum number of digits is n with a maximum of d digits to the right of the decimal point.
DATE	Date in format YYYY-MM-DD
DATETIME	Date and time in format YYYY-MM-DD HH:MI:SS
TIME	Time in format HH:MI:SS
BOOLEAN	True or False
BLOB	Binary Large Object (e.g. image, audio, video)

Key DDL Concepts

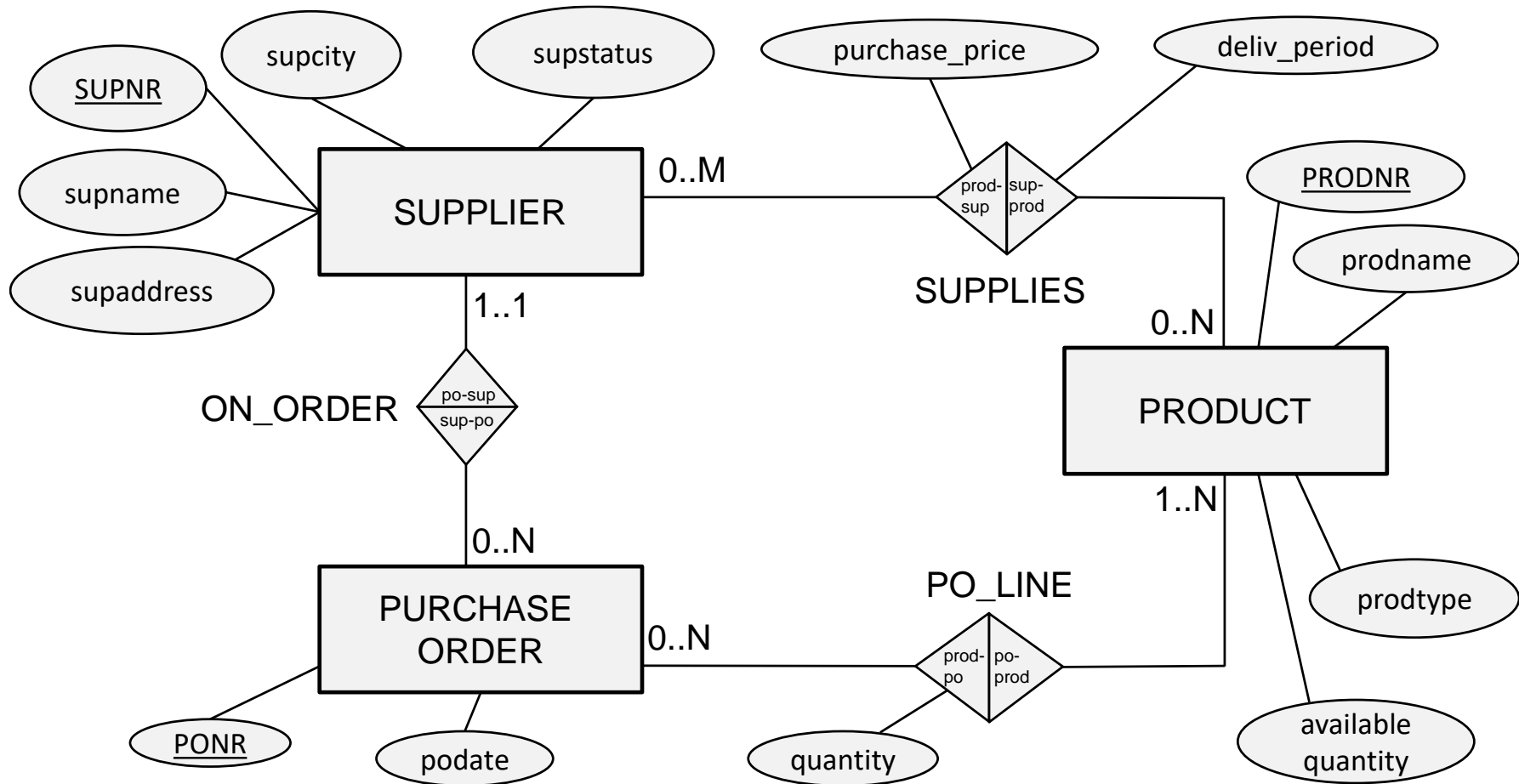
```
CREATE DOMAIN PRODTYPE_DOMAIN AS  
VARCHAR(10)
```

```
CHECK (VALUE IN ('white', 'red', 'rose',  
'sparkling'))
```

Key DDL Concepts

- Column constraints
 - **PRIMARY KEY** constraint defines the primary key of the table
 - **FOREIGN KEY** constraint defines a foreign key of a table
 - **UNIQUE** constraint defines an alternative key of a table
 - **NOT NULL** constraint prohibits NULL values for a column
 - **DEFAULT** constraint sets a default value for a column
 - **CHECK** constraint defines a constraint on the column values

DDL Example



DDL Example

SUPPLIER(SUPNR, SUPNAME, SUPADDRESS, SUPCITY, SUPSTATUS)

PRODUCT(PRODNR, PRODNAME, PRODTYPE, AVAILABLE_QUANTITY)

SUPPLIES(SUPNR, PRODNR, PURCHASE_PRICE, DELIV_PERIOD)

PURCHASE_ORDER(PONR, PODATE, *SUPNR*)

PO_LINE(PONR, PRODNR, QUANTITY)

DDL Example

CREATE TABLE SUPPLIER

**(SUPNR CHAR(4) NOT NULL PRIMARY KEY,
SUPNAME VARCHAR(40) NOT NULL,
SUPADDRESS VARCHAR(50),
SUPCITY VARCHAR(20),
SUPSTATUS SMALLINT)**

CREATE TABLE PRODUCT

**(PRODNR CHAR(6) NOT NULL PRIMARY KEY,
PRODNAME VARCHAR(60) NOT NULL,
CONSTRAINT UC1 UNIQUE(PRODNAME),
PRODTYPE VARCHAR(10),
CONSTRAINT CC1 CHECK(PRODTYPE IN ('white', 'red', 'rose', 'sparkling')),
AVAILABLE_QUANTITY INTEGER)**

DDL Example

```
CREATE TABLE SUPPLIES  
(SUPNR CHAR(4) NOT NULL,  
  PRODNR CHAR(6) NOT NULL,  
  PURCHASE_PRICE DOUBLE(8,2)  
  COMMENT 'PURCHASE_PRICE IN EUR',  
  DELIV_PERIOD TIME  
  COMMENT 'DELIV_PERIOD IN DAYS',  
  PRIMARY KEY (SUPNR, PRODNR),  
  FOREIGN KEY (SUPNR) REFERENCES SUPPLIER (SUPNR)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
  FOREIGN KEY (PRODNR) REFERENCES PRODUCT (PRODNR)  
  ON DELETE CASCADE ON UPDATE CASCADE)
```

DDL Example

```
CREATE TABLE PURCHASE_ORDER  
(PONR CHAR(7) NOT NULL PRIMARY KEY,  
PODATE DATE,  
SUPNR CHAR(4) NOT NULL,  
FOREIGN KEY (SUPNR) REFERENCES SUPPLIER (SUPNR)  
ON DELETE CASCADE ON UPDATE CASCADE)
```

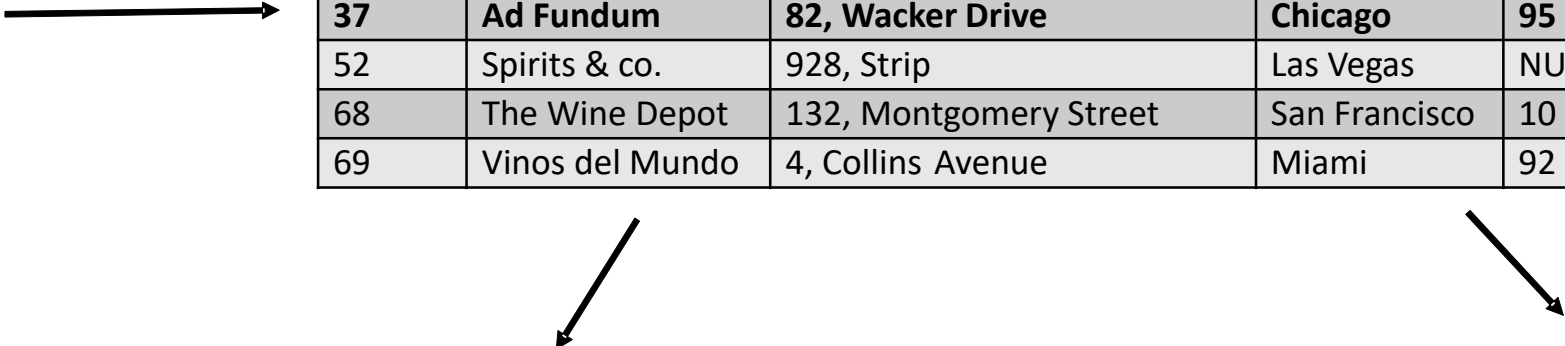
```
CREATE TABLE PO_LINE  
(PONR CHAR(7) NOT NULL,  
PRODNR CHAR(6) NOT NULL,  
QUANTITY INTEGER,  
PRIMARY KEY (PONR, PRODNR),  
FOREIGN KEY (PONR) REFERENCES PURCHASE_ORDER (PONR)  
ON DELETE CASCADE ON UPDATE CASCADE,  
FOREIGN KEY (PRODNR) REFERENCES PRODUCT (PRODNR)  
ON DELETE CASCADE ON UPDATE CASCADE)
```

Referential Integrity Constraints

- Foreign key has the same domain as the primary key it refers to and either occurs as a value of it or NULL
- What happens to foreign keys when primary key is updated or deleted?
- Options:
 - **ON UPDATE/DELETE CASCADE:** update/removal should be cascaded to all referring tuples
 - **ON UPDATE/DELETE RESTRICT:** update/removal is halted if referring tuples exist
 - **ON UPDATE/DELETE SET NULL:** foreign keys in the referring tuples are set to NULL
 - **ON UPDATE/DELETE SET DEFAULT:** foreign keys in the referring tuples are set to their default value

Referential Integrity Constraints

Supplier



<u>SUPNR</u>	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92

Supplies

<u>SUPNR</u>	<u>PRODNR</u>	PURCHASE_PRICE	DELIV_PERIOD
37	0178	16.99	4
37	0185	32.99	3
37	0468	14.00	1
37	0795	20.99	3

Purchase_Order

<u>PONR</u>	PODATE	<u>SUPNR</u>
1511	2015-03-24	37
1513	2015-04-11	37
1523	2015-04-19	37
1577	2015-05-10	37
1594	2015-05-13	37

DROP and ALTER Command

- **DROP** command can be used to drop or remove database objects
 - can be combined with CASCADE and RESTRICT
- Examples:

DROP SCHEMA PURCHASE CASCADE

DROP SCHEMA PURCHASE RESTRICT

DROP TABLE PRODUCT CASCADE

DROP TABLE PRODUCT RESTRICT

DROP and ALTER Command

- **ALTER** statement can be used to modify table column definitions
- Examples:

ALTER TABLE PRODUCT ADD PRODIMAGE BLOB

**ALTER TABLE SUPPLIER ALTER SUPSTATUS SET
DEFAULT '10'**

DROP and ALTER Command

SUPPLIER

<u>SUPNR</u>	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
...				

PRODUCT

<u>PRODNR</u>	PRODNAME	PRODTYPE	AVAILABLE_QUANTITY
0119	Chateau Miraval, Cotes de Provence Rose, 2015	rose	126
0154	Chateau Haut Brion, 2008	red	111
...		red	5

SUPPLIES

<u>SUPNR</u>	<u>PRODNR</u>	PURCHASE_PRICE	DELIV_PERIOD
21	0289	17.99	1
21	0327	56.00	6
...			

PURCHASE_ORDER

<u>PONR</u>	PODATE	SUPNR
1511	2015-03-24	37
1512	2015-04-10	94
...		

PO_LINE

<u>PONR</u>	<u>PRODNR</u>	QUANTITY
1511	0212	2
1511	0345	4
...		

SQL Data Manipulation Language (SQL DML)

- SQL SELECT Statement
- SQL INSERT Statement
- SQL DELETE Statement
- SQL UPDATE Statement

SQL SELECT Statement

- Overview
- Simple Queries
- Queries with Aggregate Functions
- Queries with GROUP BY/HAVING
- Queries with ORDER BY
- Join Queries
- Nested Queries
- Correlated Queries
- Queries with ALL/ANY
- Queries with EXISTS
- Queries with subqueries in FROM/WHERE
- Queries with Set operations

Overview

SELECT component
FROM component
[WHERE component]
[GROUP BY component]
[HAVING component]
[ORDER BY component]

Overview

- Result of SQL SELECT statement is a multiset, and not a set!
- In a multiset (bag), the elements are not ordered but there can be duplicates
- Examples: set {10, 5, 20} and multiset {10, 5, 10, 20, 5, 10}
- SQL does not eliminate duplicates
 - duplicate elimination is expensive
 - user may want to see duplicate tuples
 - duplicates may be considered by aggregate functions

Simple Queries

- SQL statements that retrieve data from only one table

**Q1: SELECT SUPNR, SUPNAME, SUPADDRESS,
SUPCITY, SUPSTATUS FROM SUPPLIER**

Q1: SELECT * FROM SUPPLIER

Simple Queries

SUPNR	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92

Simple Queries

Q2: SELECT SUPNR, SUPNAME FROM SUPPLIER

SUPNR	SUPNAME
21	Deliwines
32	Best Wines
37	Ad Fundum
52	Spirits & co.
68	The Wine Depot
69	Vinos del Mundo

Simple Queries

**Q3: SELECT SUPNR
FROM PURCHASE_ORDER**

SUPNR
32
32
37
37
37
37
37
68
69
94

Simple Queries

Q4: SELECT DISTINCT SUPNR FROM PURCHASE_ORDER

SUPNR
32
37
68
69
94

Simple Queries

**Q5: SELECT SUPNR, PRODNR, DELIV_PERIOD/30 AS
MONTH_DELIV_PERIOD FROM SUPPLIES**

SUPNR	PRODNR	MONTH_DELIV_PERIOD
21	0119	0.0333
21	0178	NULL
21	0289	0.0333
21	0327	0.2000
21	0347	0.0667
21	0384	0.0667
...

Simple Queries

**Q6: SELECT SUPNR, SUPNAME FROM SUPPLIER
WHERE SUPCITY = 'San Francisco'**

SUPNR	SUPNAME	SUPSTATUS
32	Best Wines	90
68	The Wine Depot	10

Simple Queries

**Q7: SELECT SUPNR, SUPNAME FROM SUPPLIER
WHERE SUPCITY = 'San Francisco' AND
SUPSTATUS > 80**

SUPNR	SUPNAME	SUPSTATUS
32	Best Wines	90

Simple Queries

**Q8: SELECT SUPNR, SUPNAME, SUPSTATUS
FROM SUPPLIER WHERE SUPSTATUS BETWEEN
70 AND 80**

SUPNR	SUPNAME	SUPSTATUS
94	The Wine Crate	75

Simple Queries

**Q9: SELECT PRODNR, PRODNAME
FROM PRODUCT
WHERE PRODTYPE IN ('WHITE', 'SPARKLING')**

PRODNR	PRODNAME
0178	Meerdael, Methode Traditionnelle Chardonnay, 2014
0199	Jacques Selosse, Brut Initial, 2012
0212	Billecart-Salmon, Brut Réserve, 2014
0300	Chateau des Rontets, Chardonnay, Birbettes
0494	Veuve-Cliquot, Brut, 2012
0632	Meneghetti, Chardonnay, 2010
...

Simple Queries

**Q10: SELECT PRODNR, PRODNAME
FROM PRODUCT
WHERE PRODNAME LIKE '%CHARD%'**

PRODNR	PRODNAME
0300	Chateau des Rontets, Chardonnay, Birbettes
0783	Clos D'Opleeuw, Chardonnay, 2012
0178	Meerdael, Methode Traditionnelle Chardonnay, 2014

Note: underscore (_) is a substitute for a single character!

Simple Queries

**Q11: SELECT SUPNR, SUPNAME, SUPSTATUS
FROM SUPPLIER
WHERE SUPSTATUS IS NULL**

SUPNR	SUPNAME	SUPSTATUS
52	Spirits & Co.	NULL

Queries with Aggregate Functions

- Examples: COUNT, SUM, AVG, VARIANCE, MIN/MAX, and STDEV

<u>SUPNR</u>	<u>PRODNR</u>	PURCHASE_PRICE	DELIV_PERIOD
...			
21	0178	NULL	NULL
37	0178	16.99	4
68	0178	17.99	5
69	0178	16.99	NULL
94	0178	18.00	6
...			

Queries with Aggregate Functions

**Q12: SELECT COUNT(*)
FROM SUPPLIES
WHERE PRODNR = '0178'**

5

**Q13: SELECT COUNT(PURCHASE_PRICE)
FROM SUPPLIES
WHERE PRODNR = '0178'**

4

**Q14: SELECT COUNT(DISTINCT PURCHASE_PRICE)
FROM SUPPLIES
WHERE PRODNR = '0178'**

3

Queries with Aggregate Functions

**Q15: SELECT PRODNR, SUM(QUANTITY) AS
SUM_ORDERS FROM PO_LINE
WHERE PRODNR = '0178'**

PON	PRODN	QUANTIT
R	R	Y
...		
1512	0178	3
1538	0178	6
...	0178 9	

Queries with Aggregate Functions

**Q16: SELECT SUM(QUANTITY) AS
TOTAL_ORDERS FROM PO_LINE**

173

Queries with Aggregate Functions

**Q17: SELECT PRODNR, AVG(PURCHASE_PRICE) AS
WEIGHTED_AVG_PRICE FROM SUPPLIES
WHERE PRODNR = '0178'**

<u>SUPNR</u>	<u>PRODNR</u>	PURCHASE_PRICE	DELIV_PERIOD
...			
21	0178	NULL	NULL
37	0178	16.99	4
68	0178	17.99	5
69	0178	16.99	NULL
94	0178	18.00	6
...			

0178, (16.99+17.99+16.99+18.00)/4 = 17.4925

Queries with Aggregate Functions

**Q18: SELECT PRODNR, AVG(DISTINCT
PURCHASE_PRICE)AS UNWEIGHTED_AVG_PRICE
FROM SUPPLIES WHERE PRODNR = '0178'**

<u>SUPNR</u>	<u>PRODNR</u>	PURCHASE_PRICE	DELIV_PERIOD
...			
21	0178	NULL	NULL
37	0178	16.99	4
68	0178	17.99	5
69	0178	16.99	NULL
94	0178	18.00	6
...			

0178, (16.99+17.99+18.00)/3 = 17.66

Queries with Aggregate Functions

- **Q19: SELECT PRODNR, VARIANCE(PURCHASE_PRICE)
AS PRICE_VARIANCE FROM SUPPLIES
WHERE PRODNR = '0178'**

PRODNR	PRICE_VARIANCE
0178	0.252518750000000024

Queries with Aggregate Functions

**Q20: SELECT PRODNR, MIN(PURCHASE_PRICE) AS LOWEST_PRICE,
MAX(PURCHASE_PRICE) AS HIGHEST_PRICE
FROM SUPPLIES
WHERE PRODNR = '0178'**

PRODNR	LOWEST_PRICE	HIGHEST_PRICE
0178	16.99	18.00

Queries with GROUP BY/HAVING

**Q21: SELECT PRODNR
FROM PO_LINE
GROUP BY PRODNR
HAVING COUNT(*) >= 3**

PONR	PRODNR	QUANTITY
1511	0212	2
1512	0178	3
1513	0668	7
1514	0185	2
1514	0900	2
1523	0900	3
1538	0178	6
1538	0212	15
1560	0900	9
1577	0212	6
1577	0668	9
...

Queries with GROUP BY/HAVING

GROUP BY

PONR	PRODNR	QUANTITY
1511	0212	2
1577	0212	6
1538	0212	15

PONR	PRODNR	QUANTITY
1512	0178	3
1538	0178	6

PONR	PRODNR	QUANTITY
1513	0668	7
1577	0668	9

PONR	PRODNR	QUANTITY
1514	0900	2
1523	0900	3
1560	0900	9

PONR	PRODNR	QUANTITY
1514	0185	2

PRODNR
0212
900

Queries with GROUP BY/HAVING

**Q22: SELECT PRODNR, SUM(QUANTITY) AS QUANTITY FROM PO_LINE
GROUP BY PRODNR
HAVING SUM(QUANTITY) > 15**

GROUP BY

PONR	PRODNR	QUANTITY
1511	0212	2
1577	0212	6
1538	0212	15
	<u>SUM</u>	<u>23</u>

PONR	PRODNR	QUANTITY
1514	0185	2
	<u>SUM</u>	<u>2</u>

<u>PONR</u>	PRODNR	QUANTITY
1512	0178	3
1538	0178	6
	<u>SUM</u>	<u>9</u>

<u>PONR</u>	PRODNR	QUANTITY
1514	0900	2
1523	0900	3
1560	0900	9
	<u>SUM</u>	<u>14</u>

PONR	PRODNR	QUANTITY
1513	0668	7
1577	0668	9
	<u>SUM</u>	<u>16</u>

PRODNR	QUANTITY
0212	23
0668	16

Queries with ORDER BY

**Q23: SELECT PONR, PODATE, SUPNR
FROM PURCHASE_ORDER
ORDER BY PODATE ASC, SUPNR DESC**

PONR	PODATE	SUPNR
1511	2015-03-24	37
1512	2015-04-10	94
1513	2015-04-11	37
1514	2015-04-12	32
...		

Queries with ORDER BY

**Q24: SELECT PRODNR, SUPNR, PURCHASE_PRICE
FROM SUPPLIES
WHERE PRODNR = '0178'
ORDER BY 3 DESC**

PRODNR	SUPNR	PURCHASE_PRICE
0178	94	18.00
0178	68	17.99
0178	37	16.99
0178	69	16.99
0178	21	NULL

Join Queries

- Inner Joins
- Outer Joins

Inner Joins

SUPPLIER(SUPNR, SUPNAME, ..., SUPSTATUS)
SUPPLIES(SUPNR, PRODNR, PURCHASE_PRICE, ...)

<u>SUPNR</u>	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
32	Best wines			90
68	The Wine Depot			10
84	Wine Trade Logistics			92
:	:			:

<u>SUPNR</u>	<u>PRODNR</u>	PURCHASE_PRICE	DELIV_PERIOD
32	0474	40.00	1
32	0154	21.00	4
84	0494	15.99	2
:	:	:	

Inner Joins

**Q25: SELECT R.SUPNR, R.SUPNAME, R.SUPSTATUS,
S.SUPNR, S.PRODNR, S.PURCHASE_PRICE
FROM SUPPLIER R, SUPPLIES S**

Inner Joins

R.SUPNR	R.SUPNAME	R.SUPSTATUS	S.SUPNR	S.PRODNR	S.PURCHASE_PRICE
21	Deliwines	20	21	0119	15.99
32	Best Wines	90	21	0119	15.99
37	Ad Fundum	95	21	0119	15.99
52	Spirits & co.	NULL	21	0119	15.99
...					
32	Best Wines	90	32	0154	21.00
37	Ad Fundum	95	32	0154	21.00
52	Spirits & co.	NULL	32	0154	21.00
...					
69	Vinos del Mundo	92	94	0899	15.00
84	Wine Trade Logistics	92	94	0899	15.00
94	The Wine Crate	75	94	0899	15.00

Inner Joins

**Q26: SELECT R.SUPNR, R.SUPNAME,
R.SUPSTATUS, S.PRODNR,
S.PURCHASE_PRICE
FROM SUPPLIER R, SUPPLIES S
WHERE R.SUPNR = S.SUPNR**

Inner Joins

R.SUPNR	R.SUPNAME	R.SUPSTATUS	S.SUPNR	S.PRODNR	S.PURCHASE_PRICE
21	Deliwines	20	21	0119	15.99
21	Deliwines	20	21	0178	NULL
21	Deliwines	20	21	0289	17.99
21	Deliwines	20	21	0327	56.00
21	Deliwines	20	21	0347	16.00
21	Deliwines	20	21	0384	55.00
21	Deliwines	20	21	0386	58.99
21	Deliwines	20	21	0468	14.99
21	Deliwines	20	21	0668	6.00
32	Best Wines	90	32	0154	21.00
32	Best Wines	90	32	0474	40.00
32	Best Wines	90	32	0494	15.00
32	Best Wines	90	32	0657	44.99
32	Best Wines	90	32	0760	52.00
...					

Inner Joins

**Q27: SELECT R.SUPNR, R.SUPNAME, R.SUPSTATUS,
S.PRODNR, S.PURCHASE_PRICE
FROM SUPPLIER AS R INNER JOIN SUPPLIES AS S
ON (R.SUPNR = S.SUPNR)**

Inner Joins

**Q28: SELECT R.SUPNR, R.SUPNAME, PO.PONR, PO.PODATE,
P.PRODNR,P.PRODNAME, POL.QUANTITY
FROM SUPPLIER R, PURCHASE_ORDER PO, PO_LINE
POL, PRODUCT P
WHERE (R.SUPNR = PO.SUPNR)
AND (PO.PONR = POL.PONR)
AND (POL.PRODNR = P.PRODNR)**

Inner Joins

R.SUPNR	R.SUPNAME	PO.PONR	PO.PODATE	P.PRODNR	P.PRODNAME	POL.QUANTITY
37	Ad Fundum	1511	2015-03-24	0212	Billecart-Salmon, Brut Réserve, 2014	2
37	Ad Fundum	1511	2015-03-24	0345	Vascosassetti, Brunello di Montalcino, 2004	4
37	Ad Fundum	1511	2015-03-24	0783	Clos D'Opleeuw, Chardonnay, 2012	1
37	Ad Fundum	1511	2015-03-24	0856	Domaine Chandon de Briailles, Savigny-Les-Beaune, 2006	9
94	The Wine Crate	1512	2015-04-10	0178	Meerdael, Methode Traditionnelle Chardonnay, 2014	3
...						

Inner Joins

**Q29: SELECT R1.SUPNAME, R2.SUPNAME,
R1.SUPCITY
FROM SUPPLIER R1, SUPPLIER R2
WHERE R1.SUPCITY = R2.SUPCITY
AND (R1.SUPNR < R2.SUPNR)**

Inner Joins

<u>SUPNR</u>	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
21	Deliwines	240, Avenue of the Americas	New York	20
32	Best Wines	660, Market Street	San Francisco	90
37	Ad Fundum	82, Wacker Drive	Chicago	95
52	Spirits & co.	928, Strip	Las Vegas	NULL
68	The Wine Depot	132, Montgomery Street	San Francisco	10
69	Vinos del Mundo	4, Collins Avenue	Miami	92

SUPNAME	SUPNAME	SUPCITY
Best Wines	The Wine Depot	San Francisco

Inner Joins

**Q30: SELECT R.SUPNAME
FROM SUPPLIER R, SUPPLIES S
WHERE R.SUPNR = S.SUPNR
AND S.PRODNR = '0899'**

Wine Crate

Inner Joins

**Q31: SELECT DISTINCT R.SUPNAME
FROM SUPPLIER R, SUPPLIES S, PRODUCT P
WHERE S.SUPNR = R.SUPNR
AND S.PRODNR = P.PRODNR
AND P.PRODTYPE = 'ROSE'**

SUPNAME
DeliWines
DeliWines
DeliWines
The Wine Depot

SUPNAME
DeliWines
The Wine Depot

Inner Joins

**Q32: SELECT P.PRODNR, P.PRODNAME, SUM(POL.QUANTITY)
FROM PRODUCT P, PO_LINE POL
WHERE P.PRODNR = POL.PRODNR
GROUP BY P.PRODNR**

PRODNR	PRODNAME	SUM(POL.QUANTITY)
0178	Meerdael, Methode Traditionnelle Chardonnay, 2014	9
0185	Chateau Petrus, 1975	2
0212	Billecart-Salmon, Brut Réserve, 2014	23
0295	Chateau Pape Clement, Pessac-Léognan, 2001	9
0306	Chateau Coupe Roses, Granaxa, 2011	11
...		

Outer Joins

- Outer join can be used when we want to keep all the tuples of one, or both tables, in the result of the JOIN, regardless of whether or not they have matching tuples in the other table

Outer Joins

**Q33: SELECT R.SUPNR, R.SUPNAME, R.SUPSTATUS,
S.PRODNR, S.PURCHASE_PRICE
FROM SUPPLIER AS R LEFT OUTER JOIN SUPPLIES AS S
ON (R.SUPNR = S.SUPNR)**

<u>SUPNR</u>	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS
68	The Wine Depot			
21	Deliwines			
94	The Wine Crate			
..				

<u>SUPNR</u>	<u>PRODNR</u>	PURCHASE_PRICE	DELIV_PERIOD
21	0119	15.99	1
21	0289	17.99	1
68	0178	17.99	5
:	:	:	:

Outer Joins

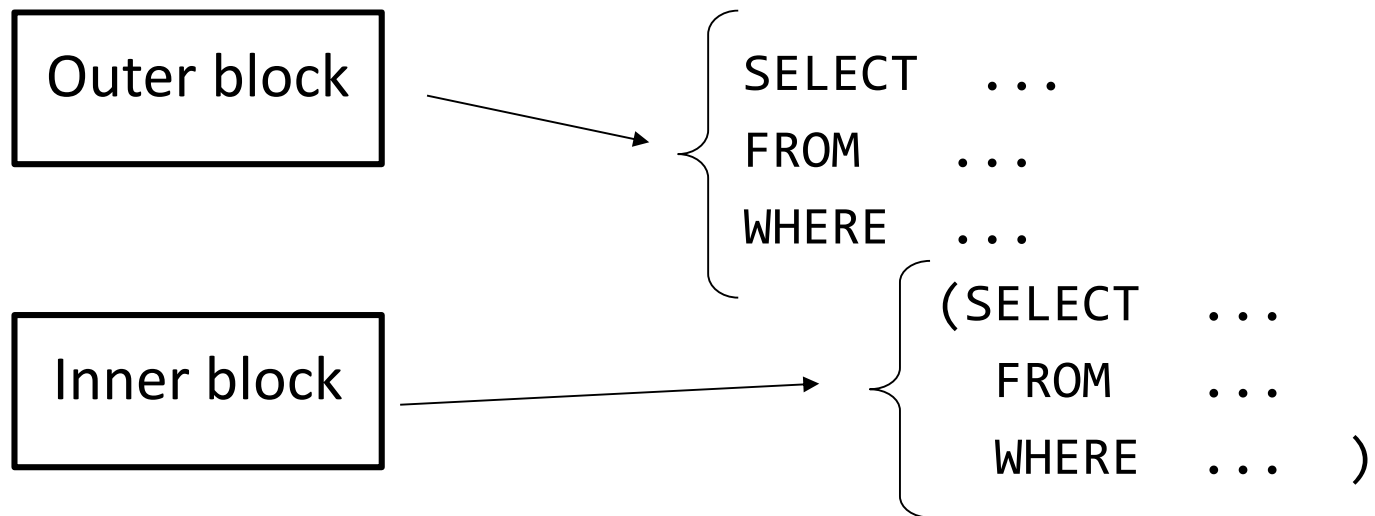
<u>SUPNR</u>	SUPNAME	SUPSTATUS	PRODNR	PURCHASE_PRICE
21	Deliwines	20	0119	15.99
21	Deliwines	20	0178	NULL
...				
37	Ad Fundum	95	0795	20.99
52	Spirits & Co.	NULL	NULL	NULL
68	The Wine Depot	10	0178	17.99
...				

Outer Joins

**Q34: SELECT P.PRODNR, P.PRODNAME, SUM(POL.QUANTITY)
AS SUM FROM PO_LINE AS POL RIGHT OUTER JOIN PRODUCT AS P
ON (POL.PRODNR = P.PRODNR)
GROUP BY P.PRODNR**

P.PRODNR	P.PRODNAME	SUM
0119	Chateau Miraval, Cotes de Provence Rose, 2015	NULL
0154	Chateau Haut Brion, 2008	NULL
0178	Meerdael, Methode Traditionnelle Chardonnay, 2014	9
0185	Chateau Petrus, 1975	2
0199	Jacques Selosse, Brut Initial, 2012	NULL
0212	Billecart-Salmon, Brut Réserve, 2014	23
...		

Nested Queries



Nested Queries

**Q34: SELECT SUPNAME
FROM SUPPLIER
WHERE SUPNR =
(SELECT SUPNR
FROM PURCHASE_ORDER
WHERE PONR = '1560')**

Best Wines

Nested Queries

**Q35: SELECT PRODNR, PRODNAME
FROM PRODUCT
WHERE AVAILABLE_QUANTITY >
(SELECT AVAILABLE_QUANTITY
FROM PRODUCT
WHERE PRODNR = '0178')**

PRODNR	PRODNAME
0212	Billecart-Salmon, Brut Réserve, 2014
0347	Chateau Corbin-Despaigne, Saint-Emilion, 2005
0474	Chateau De La Tour, Clos-Vougeot, Grand cru, 2008
0885	Chateau Margaux, Grand Cru Classé, 1956
0899	Trimbach, Riesling, 1989

Nested Queries

```
Q36: SELECT SUPNAME  
FROM SUPPLIER  
WHERE SUPNR IN  
      (SELECT SUPNR  
        FROM SUPPLIES  
        WHERE PRODNR = '0178' )
```

SUPNAME
Deliwines
Ad Fundum
The Wine Depot
Vinos del Mundo
The Wine Crate

Nested Queries

**Q37: SELECT SUPNAME
FROM SUPPLIER
WHERE SUPNR IN
(SELECT SUPNR
FROM SUPPLIES
WHERE PRODNR IN
(SELECT PRODNR
FROM PRODUCT
WHERE PRODTYPE = 'ROSE'))**

SUPNAME
Deliwines
The Wine Depot

Nested Queries

**Q38: SELECT PRODNAME
FROM PRODUCT
WHERE PRODNR IN
(SELECT PRODNR
FROM SUPPLIES
WHERE SUPNR = '32')
AND PRODNR IN
(SELECT PRODNR
FROM SUPPLIES
WHERE SUPNR = '84')**

PRODNAME
Veuve-Cliquot, Brut, 2012
Conde de Hervías, Rioja, 2004

Correlated Queries

- Whenever a condition in the WHERE clause of a nested query references some column of a table declared in the outer query, the two queries are said to be correlated
- The nested query is then evaluated once for each tuple (or combination of tuples) in the outer query

Correlated Queries

**Q39: SELECT P.PRODNR
FROM PRODUCT P
WHERE 1 <
(SELECT COUNT(*)
FROM PO_LINE POL
WHERE P.PRODNR = POL.PRODNR)**

PRODNR	PRODNAME
0212	Billecart-Salmon, Brut Réserve, 2014
0289	Chateau Saint Estève de Neri, 2015
0154	Chateau Haut Brion, 2008
0295	Chateau Pape Clement, Pessac-Léognan, 2001
...	

PONR	PRODNR	QUANTITY
...		
1511	0212	2
...		
1538	0212	15
...		
1577	0212	6
...		

Correlated Queries

Q40: **SELECT** R.SUPNR, R.SUPNAME, P.PRODNR, P.PRODNAME,
S1.PURCHASE_PRICE, S1.DELIV_PERIOD
FROM SUPPLIER R, SUPPLIES S1, PRODUCT P
WHERE R.SUPNR = S1.SUPNR
AND S1.PRODNR = P.PRODNR
AND S1.PURCHASE_PRICE <
 (**SELECT** **AVG**(PURCHASE_PRICE)
 FROM SUPPLIES S2
 WHERE P.PRODNR = S2.PRODNR)

Correlated Queries

PRODUCT P

<u>PRODNR</u>	PRODNAME	PRODTYPE
0178	Meerdael , Methode Traditionnelle Chardonnay, 2014	sparkling
0185	Chateau Petrus, 1975	red
..		

SUPPLIES S1

<u>PRODNR</u>	<u>SUPNR</u>	PURCHASE_PRICE	DELIV_PERIOD
0178	37	16.99	4
0178	68	17.99	5
0178	69	16.99	-
0178	94	18.00	6
0178	21	-	-
0185	37	32.99	3
...			

SUPPLIER R

<u>SUPNR</u>	SUPNAME	SUPADDRESS
37	Ad Fundum	...
68	The Wine Depot	...
84	Wine Trade Logistics	...
...		

SUPPLIES S2

<u>PRODNR</u>	<u>SUPNR</u>	PURCHASE_PRICE	DELIV_PERIOD
0178	37	16.99	4
0178	68	17.99	5
0178	69	16.99	-
0178	94	18.00	6
0178	21	-	-
0185	37	32.99	3
...			

< AVG ?

Correlated Queries

**Q41: SELECT P1.PRODNR
FROM PRODUCT P1
WHERE 3 >**

**(SELECT COUNT(*)
FROM PRODUCT P2
WHERE P1.PRODNR < P2.PRODNR)**

P1.PRODNR	Result of Inner Query block	< 3?	Output
0119	41	No	No
0154	40	No	No
0178	39	No	No
...
0899	3	No	No
0900	2	Yes	Yes
0915	1	Yes	Yes
0977	0	Yes	Yes

Queries with ALL/ANY

- The comparison condition $v > \text{ALL } V$ returns TRUE if the value v is greater than all the values in the multiset V
 - If the nested query doesn't return a value, it evaluates the condition as TRUE
- The comparison condition $v > \text{ANY } V$ returns TRUE if the value v is greater than at least one value in the multiset V
 - If the nested query doesn't return a value, it evaluates the whole condition as FALSE

Queries with ALL/ANY

```
Q42: SELECT SUPNAME
      FROM SUPPLIER
      WHERE SUPNR IN
        (SELECT SUPNR
         FROM SUPPLIES
         WHERE PRODNR = '0668'
         AND PURCHASE_PRICE >= ALL
          (SELECT PURCHASE_PRICE
           FROM SUPPLIES
           WHERE PRODNR = '0668'))
```

The Wine Depot

68

6.00, 6.99

Queries with ALL/ANY

PRODUCT P

<u>PRODNR</u>	PRODNAME	PRODTYPE
0178	Meerdael, Methode Traditionnelle Chardonnay, 2014	sparkling
0668	Gallo Family Vineyards, Grenache, 2014	rose
...		

SUPPLIES S1

<u>PRODNR</u>	<u>SUPNR</u>	PURCHASE_PRICE	DELIV_PERIOD
0668	68	6.99	3
0668	21	6.00	1
0760	32	52.00	3
0760	68	52.99	2
0783	69	7.00	3
...

SUPPLIER R

<u>SUPNR</u>	SUPNAME	SUPADDRESS
32	Best wines	...
68	The Wine Depot	...
84	Wine Trade Logistics	...
..		

SUPPLIES S2

<u>PRODNR</u>	<u>SUPNR</u>	PURCHASE_PRICE	DELIV_PERIOD
0668	68	6.99	3
0668	21	6.00	1
0760	32	52.00	3
0760	68	52.99	2
0783	69	7.00	3
...

? ALL ?

Queries with ALL/ANY

**Q43: SELECT R1.SUPNR, R1.SUPNAME, R1.SUPCITY, R1.SUPSTATUS
FROM SUPPLIER R1
WHERE R1.SUPSTATUS >= ALL
(SELECT R2.SUPSTATUS
FROM SUPPLIER R2
WHERE R1.SUPCITY = R2.SUPCITY)**

SUPNR	SUPNAME	SUPCITY	SUPSTATUS
21	Deliwines	New York	20
32	Best Wines	San Francisco	90
37	Ad Fundum	Chicago	95
69	Vinos del Mundo	Miami	92
84	Wine Trade Logistics	Washington	92
94	The Wine Crate	Dallas	75

Queries with ALL/ANY

**Q44: SELECT SUPNAME
FROM SUPPLIER
WHERE SUPNR IN**

The Wine Depot, The Wine Crate

**(SELECT SUPNR
FROM SUPPLIES
WHERE PRODNR = '0178' AND
PURCHASE_PRICE > ANY**

68,94

**(SELECT PURCHASE_PRICE
FROM SUPPLIES
WHERE PRODNR = '0178'))**

NULL, 16.99, 17.99, 16.99, 18.00

Queries with EXISTS

- EXISTS function checks whether the result of a correlated nested query is empty or not
- The result is a Boolean value: TRUE or FALSE
- EXISTS returns TRUE if there is at least one tuple in the result of the nested query, or otherwise returns FALSE
- Vice versa, the NOT EXISTS function returns TRUE if there are no tuples in the result of the nested query, or otherwise returns FALSE

Queries with EXISTS

**Q44: SELECT SUPNAME
FROM SUPPLIER R
WHERE EXISTS
(SELECT *
FROM SUPPLIES S
WHERE R.SUPNR = S.SUPNR
AND S.PRODNR = '0178')**

SUPNAME
Deliwines
Ad Fundum
The Wine Depot
Vinos del Mundo
The Wine Crate

Queries with EXISTS

**Q45: SELECT SUPNAME, SUPADDRESS, SUPCITY
FROM SUPPLIER R
WHERE NOT EXISTS
 (SELECT *
 FROM PRODUCT P
 WHERE NOT EXISTS
 (SELECT *
 FROM SUPPLIES S
 WHERE R.SUPNR = S.SUPNR
 AND P.PRODNR = S.PRODNR))**

Queries with Subqueries in SELECT/FROM

**Q46: SELECT P.PRODNR, P.PRODNAME,
(SELECT SUM(QUANTITY) FROM PO_LINE POL
WHERE P.PRODNR =POL.PRODNR) AS TOTALORDERED
FROM PRODUCT P**

PRODNR	PRODNAME	TOTALORDERED
0212	Billecart-Salmon, Brut Réserve, 2014	23
0795	Casa Silva, Los Lingues, Carmenere, 2012	3
0915	Champagne Boizel, Brut, Réserve, 2010	13
0523	Chateau Andron Blanquet, Saint Estephe, 1979	NULL
0977	Chateau Batailley, Grand Cru Classé, 1975	11
...		

Queries with Subqueries in SELECT/FROM

**Q47: SELECT M.PRODNR, M.MINPRICE, M.MAXPRICE FROM
(SELECT PRODNR, MIN(PURCHASE_PRICE) AS MINPRICE,
MAX(PURCHASE_PRICE) AS MAXPRICE
FROM SUPPLIES GROUP BY PRODNR) AS M
WHERE M.MAXPRICE-M.MINPRICE > 1**

PRODNR	MINPRICE	MAXPRICE
0178	16.99	18.00
0199	30.99	32.00
0300	19.00	21.00
0347	16.00	18.00
0468	14.00	15.99

Queries with Set Operations

- $A = \{10, 5, 25, 30, 45\}$
- $B = \{15, 20, 10, 30, 50\}$
- $A \text{ UNION } B = \{5, 10, 15, 20, 25, 30, 45, 50\}$
- $A \text{ INTERSECT } B = \{10, 30\}$
- $A \text{ EXCEPT } B = \{5, 25, 45\}$

Queries with Set Operations

**Q48: SELECT SUPNR, SUPNAME
FROM SUPPLIER
WHERE SUPCITY = 'New York'
UNION
SELECT R.SUPNR, R.SUPNAME
FROM SUPPLIER R, SUPPLIES S
WHERE R.SUPNR = S.SUPNR
AND S.PRODNR = '0915'
ORDER BY SUPNAME ASC**

SUPNR	SUPNAME
21	Deliwines
84	Wine Trade Logistics

Queries with Set Operations

**Q49: SELECT SUPNR, SUPNAME
FROM SUPPLIER
WHERE SUPCITY = 'NEW YORK'
INTERSECT
SELECT R.SUPNR, R.SUPNAME
FROM SUPPLIER R, SUPPLIES S
WHERE R.SUPNR = S.SUPNR
AND S.PRODNR = '0915'
ORDER BY SUPNAME ASC**

NULL

Queries with Set Operations

**Q50: SELECT SUPNR
FROM SUPPLIER
EXCEPT
SELECT SUPNR
FROM SUPPLIES**

52

SQL INSERT Statement

INSERT INTO PRODUCT VALUES

('980', 'Chateau Angelus, Grand Clu Classé, 1960', 'red', 6)

**INSERT INTO PRODUCT(PRODNR, PRODNAME, PRODTYPE,
AVAILABLE_QUANTITY) VALUES**

('980', 'Chateau Angelus, Grand Clu Classé, 1960', 'red', 6)

INSERT INTO PRODUCT(PRODNR, PRODNAME, PRODTYPE) VALUES

('980', 'Chateau Angelus, Grand Clu Classé, 1960', 'red')

SQL INSERT Statement

```
INSERT INTO PRODUCT(PRODNR, PRODNAME, PRODTYPE,  
AVAILABLE_QUANTITY) VALUES  
( '980', 'Chateau Angelus, Grand Clu Classé, 1960', 'red', 6),  
( '1000', 'Domaine de la Vougeraie, Bâtard Montrachet', Grand  
cru, 2010', 'white', 2),  
( '1002', 'Leeuwin Estate Cabernet Sauvignon 2011', 'white',  
20)
```

```
INSERT INTO INACTIVE-SUPPLIERS(SUPNR)  
SELECT SUPNR  
FROM SUPPLIER  
EXCEPT  
SELECT SUPNR  
FROM SUPPLIES
```

SQL DELETE Statement

```
DELETE FROM PRODUCT  
WHERE PRODNR = '1000'
```

```
DELETE FROM SUPPLIER  
WHERE SUPSTATUS IS NULL
```

```
DELETE FROM SUPPLIES  
WHERE PRODNR IN (SELECT PRODNR  
                  FROM PRODUCT  
                  WHERE PRODNAME LIKE '%CHARD%')
```

SQL DELETE Statement

```
DELETE FROM SUPPLIER R
WHERE NOT EXISTS
    (SELECT PRODNR
     FROM SUPPLIES S
     WHERE R.SUPNR=S.SUPNR)
```

```
DELETE FROM SUPPLIES S1
WHERE S1.PURCHASE_PRICE >
    (SELECT 2*AVG(S2.PURCHASE_PRICE)
     FROM SUPPLIES S2
     WHERE S1.PRODNR=S2.PRODNR)
```

```
DELETE FROM PRODUCT
```

SQL UPDATE Statement

```
UPDATE PRODUCT  
SET AVAILABLE_QUANTITY=26  
WHERE PRODNR='0185'
```

```
UPDATE SUPPLIER  
SET SUPSTATUS = DEFAULT
```

```
UPDATE SUPPLIES  
SET DELIV_PERIOD= DELIV_PERIOD+7  
WHERE SUPNR IN (SELECT SUPNR  
                  FROM SUPPLIER  
                  WHERE SUPNAME = 'Deliwines')
```

SQL UPDATE Statement

```
UPDATE SUPPLIES S1
SET (PURCHASE_PRICE, DELIV_PERIOD)=
(SELECT MIN(PURCHASE_PRICE), MIN(DELIV_PERIOD)
FROM SUPPLIES S2
WHERE S1.PRODNR=S2.PRODNR)
WHERE SUPNR='68'
```

```
ALTER TABLE SUPPLIER ADD SUPCATEGORY VARCHAR(10) DEFAULT
'SILVER'
```

```
UPDATE SUPPLIER
SET SUPCATEGORY =
CASE WHEN SUPSTATUS >=70 AND SUPSTATUS <=90 THEN 'GOLD'
WHEN SUPSTATUS >=90 THEN 'PLATINUM'
ELSE 'SILVER'
END
```

SQL UPDATE Statement

<u>SUPNR</u>	SUPNAME	SUPADDRESS	SUPCITY	SUPSTATUS	SUPCATEGORY
21	Deliwines	20, Avenue of the Americas	New York	20	SILVER
32	Best Wines	660, Market Street	San Francisco	90	GOLD
37	Ad Fundum	82, Wacker Drive	Chicago	95	PLATINUM
52	Spirits & co.	928, Strip	Las Vegas	NULL	SILVER
68	The Wine Depot	132, Montgomery Street	San Francisco	10	SILVER
69	Vinos del Mundo	4, Collins Avenue	Miami	92	PLATINUM
84	Wine Trade Logistics	100, Rhode Island Avenue	Washington	92	PLATINUM
94	The Wine Crate	330, McKinney Avenue	Dallas	75	GOLD

SQL Views

- SQL views are part of the external data model
- A view is defined by means of an SQL query and its content is generated upon invocation of the view by an application or other query
- A view is a virtual table without physical tuples
- Views allow for logical data independence which makes them a key component in the three-layer database architecture

SQL Views

```
CREATE VIEW TOPSUPPLIERS  
AS SELECT SUPNR, SUPNAME FROM SUPPLIER  
WHERE SUPSTATUS > 50
```

```
CREATE VIEW TOPSUPPLIERS_SF  
AS SELECT * FROM TOPSUPPLIERS  
WHERE SUPCITY='San Francisco'
```


SQL Views

```
CREATE VIEW ORDEROVERVIEW(PRODNR,  
PRODNAME, TOTQUANTITY)  
AS SELECT P.PRODNR, P.PRODNAME,  
SUM(POL.QUANTITY)  
FROM PRODUCT AS P LEFT OUTER JOIN  
PO_LINE AS POL  
ON (P.PRODNR = POL.PRODNR)  
GROUP BY P.PRODNR
```

SQL Views

```
SELECT * FROM TOPSUPPLIERS_SF
```

```
SELECT * FROM ORDEROVERVIEW  
WHERE PRODNAME LIKE '%CHARD%'
```

SQL Views

- Query modification: RDBMS modifies queries that query views into queries on the underlying base tables
- View materialization: a physical table is created when the view is first queried

SQL Views

- Some views can be updated
 - In this case, the view serves as a window through which updates are propagated to the underlying base table(s)

SQL Views

```
CREATE VIEW ORDEROVERVIEW(PRODNR, PRODNAME,  
TOTQUANTITY)  
AS SELECT P.PRODNR, P.PRODNAME, SUM(POL.QUANTITY)  
FROM PRODUCT AS P LEFT OUTER JOIN PO_LINE AS POL  
ON (P.PRODNR = POL.PRODNR)  
GROUP BY P.PRODNR
```

```
UPDATE VIEW ORDEROVERVIEW  
SET TOTQUANTITY=10  
WHERE PRODNR= '0154'
```

ERROR!

SQL Views

- **WITH CHECK** option checks UPDATE and INSERT statements for conformity with the view definition

```
CREATE VIEW TOPSUPPLIERS  
AS SELECT SUPNR, SUPNAME FROM SUPPLIER  
WHERE SUPSTATUS > 50  
WITH CHECK OPTION
```

```
UPDATE TOPSUPPLIERS  
SET STATUS =20  
WHERE SUPNR='32'
```

OK!

```
UPDATE TOPSUPPLIERS  
SET STATUS =80  
WHERE SUPNR='32'
```

NOT OK!

SQL Privileges

- A privilege corresponds to the right to use certain SQL statements such as SELECT, INSERT, etc. on one or more database objects

Privilege	Explanation
SELECT	Provides retrieval privilege
INSERT	Gives insert privilege
UPDATE	Gives update privilege
DELETE	Gives delete privilege
ALTER	Gives privilege to change the table definition
REFERENCES	Provides the privilege to reference the table when specifying integrity constraints.
ALL	Provides all privileges (DBMS specific)

SQL Privileges

GRANT SELECT, INSERT, UPDATE, DELETE ON SUPPLIER TO BBAESENS

GRANT SELECT (PRODNR, PRODNAME) ON PRODUCT TO PUBLIC

REVOKE DELETE ON SUPPLIER FROM BBAESENS

GRANT SELECT, INSERT, UPDATE, DELETE ON PRODUCT TO WLEMAHIEU WITH GRANT OPTION

GRANT REFERENCES ON SUPPLIER TO SVANDENBROUCKE

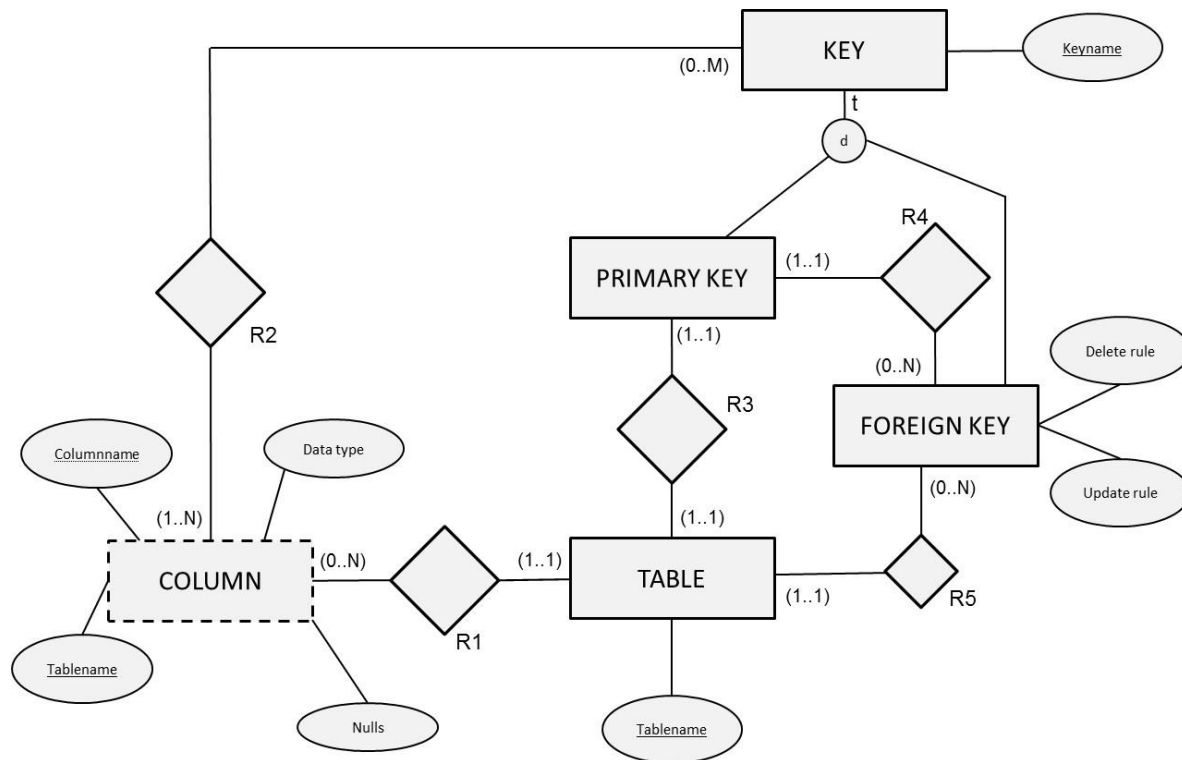
SQL Privileges

```
CREATE VIEW SUPPLIERS_NY  
AS SELECT SUPNR, SUPNAME FROM SUPPLIERS  
WHERE SUPCITY='New York'
```

```
GRANT SELECT ON SUPPLIERS_NY TO  
WLEMAHIEU
```

SQL for Metadata Management

- Catalog itself can also be implemented as a relational database



SQL for Metadata Management

Table(Tablename, ...)

Key(Keyname, ...)

Primary-Key(PK-Keyname, *PK-Tablename*, ...)

PK-Keyname is a foreign key referring to Keyname in Key

PK-Tablename is a foreign key referring to Tablename in Table

Foreign-Key(FK-Keyname, *FK-Tablename*, *FK-PK-Keyname*, Update-rule, Delete-rule, ...)

FK-Keyname is a foreign key referring to Keyname in Key

FK-Tablename is a foreign key referring to Tablename in Table

FK-PK-Keyname is a foreign key referring to PK-Keyname in Primary-Key

Column(Columnname, *C-Tablename*, Data type, Nulls, ...)

C-Tablename is a foreign key referring to Tablename in Table

Key-Column(KC-Keyname, KC-Columnname, *KC-Tablename*, ...)

KC-Keyname is a foreign key referring to Keyname in Key

KC-Columnname is a foreign key referring to Columnname in Column

KC-Tablename is a foreign key referring to C-Tablename in Column

SQL for Metadata Management

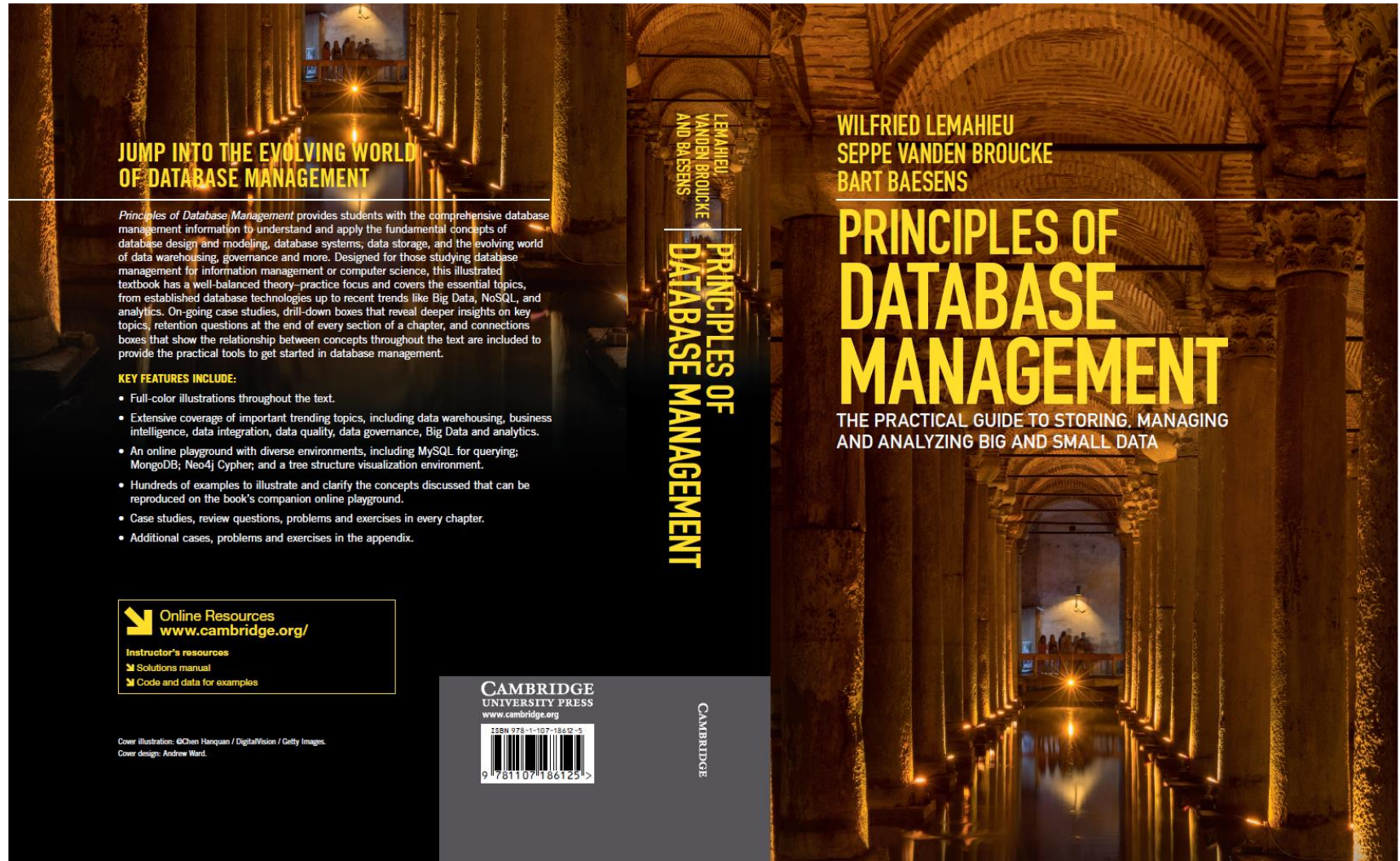
```
SELECT *  
FROM Column  
WHERE Tablename='SUPPLIER'
```

```
SELECT PK.PK-Keyname, FK.FK-PK-Keyname, FK.FK-  
Tablename, FK.Delete-rule  
FROM Primary-Key PK, Foreign-Key FK  
WHERE PK.PK-Tablename='SUPPLIER'  
AND PK.PK-Keyname= FK.FK-PK-Keyname
```

Conclusions

- Relational Database Management Systems and SQL
- SQL Data Definition Language
- SQL Data Manipulation Language
- SQL Views
- SQL Privileges
- SQL for Metadata Management

More information?



www.pdbmbook.com