Databases

(6G4Z0016)

Introduction to Databases

Stephen Gordon

E140

s.gordon@mmu.ac.uk

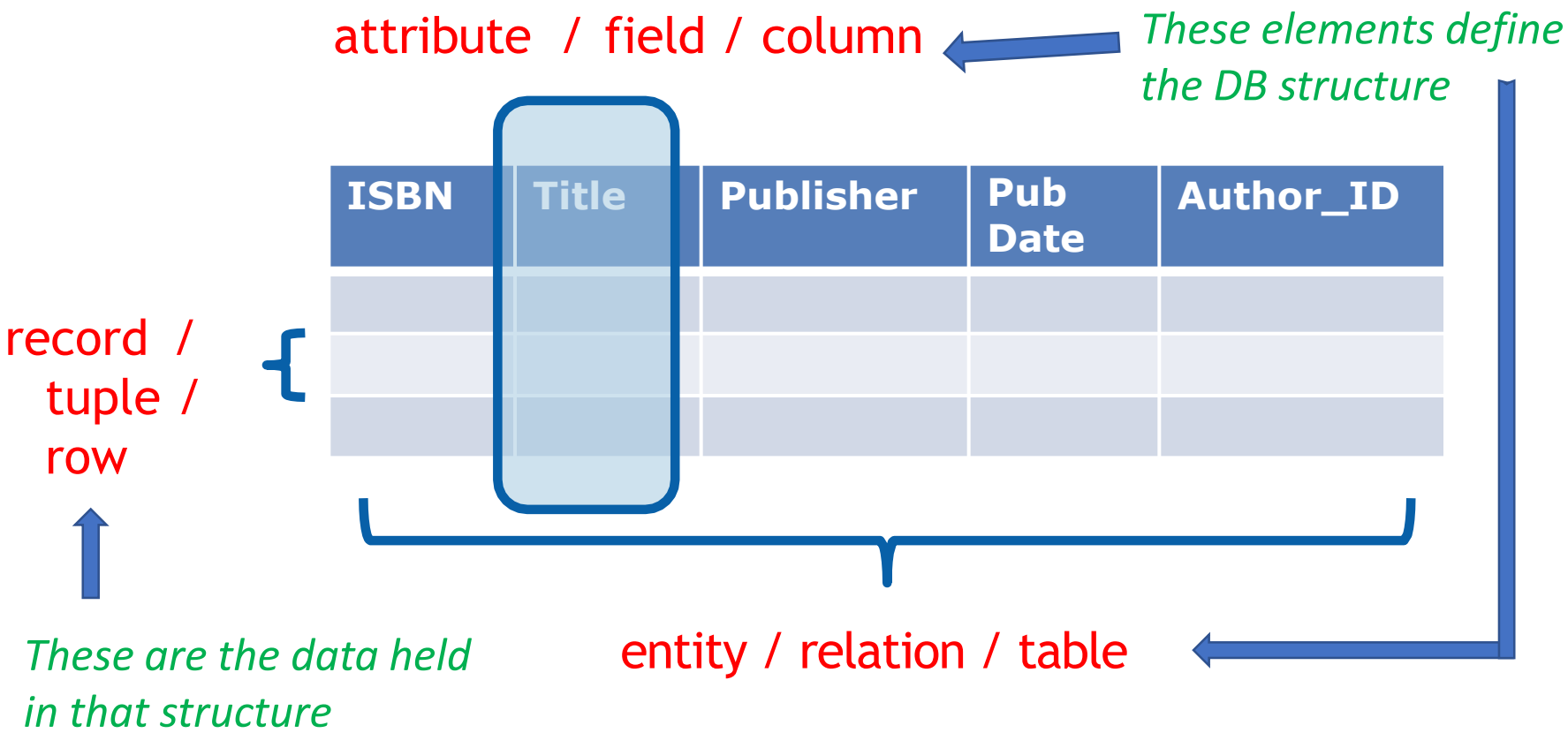
What you will learn

- Tables/entities, fields/attributes
- What is a Database? What is a DBMS?
- What is SQL?
- SELECT statement examples and structure
- WHERE statement and comparison operators
- Entity and Referential Integrity
- ANSI/SPARC architecture
- 3-tier architecture
- Business context

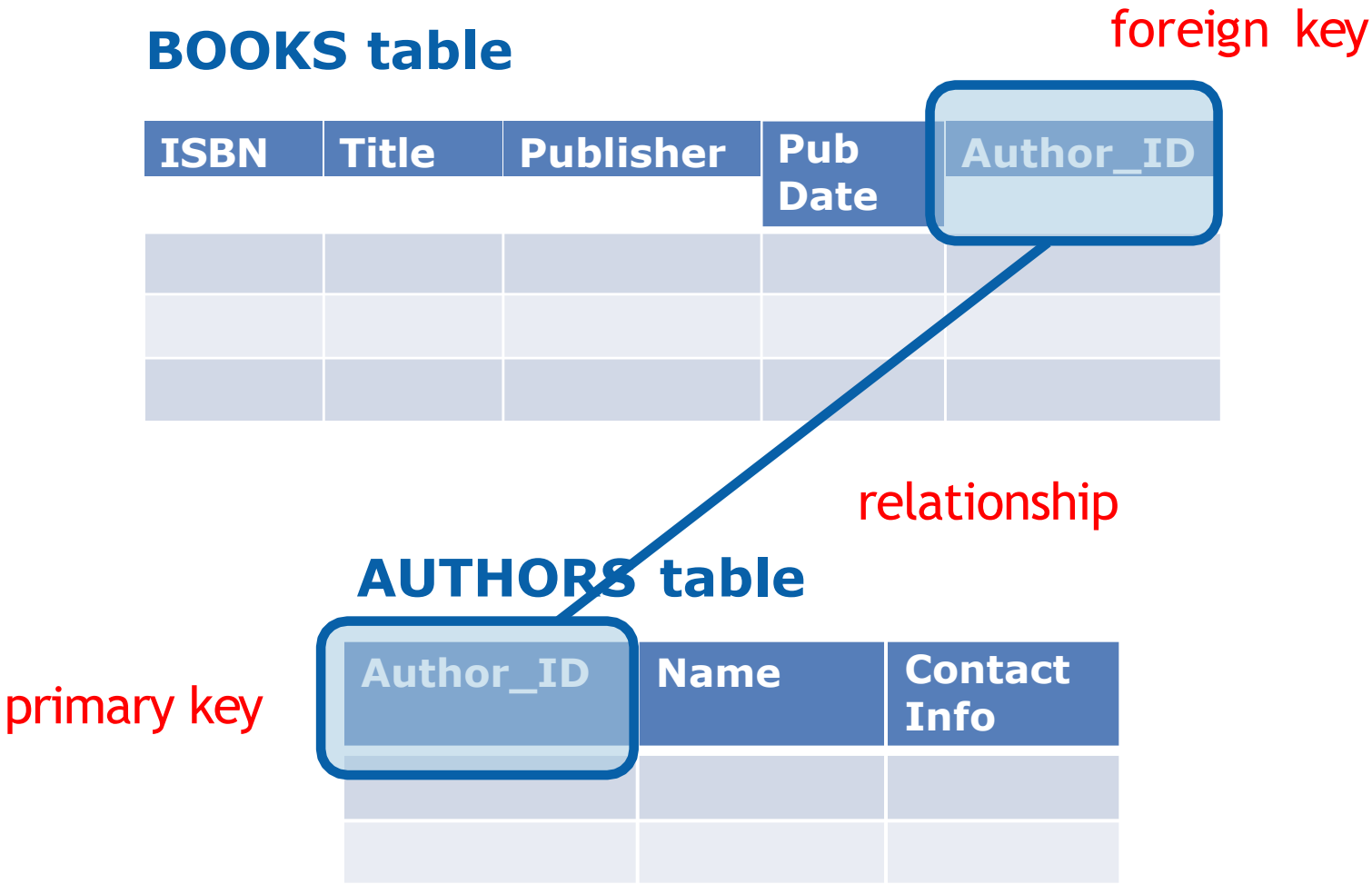
RECAP

Database Tables/Entities

Entities are nouns and can be tangible (car, book) or intangible (account, insurance policy)



Keys define relationships between tables



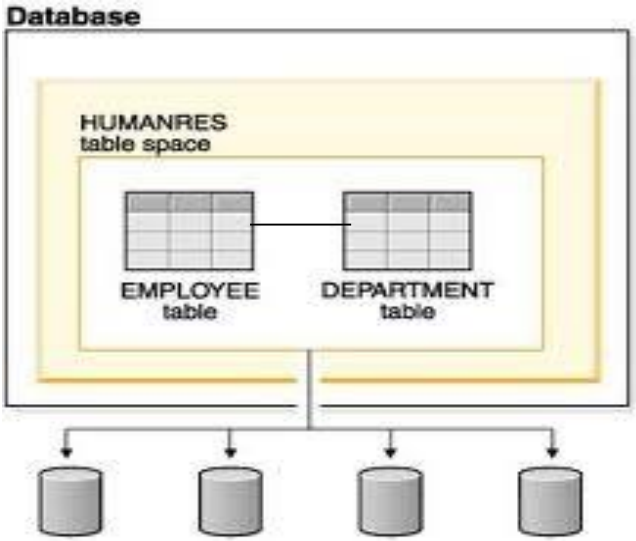
What is a database?

A database is a collection (integration) of entities (called tables)

When entities are “related”, the database is called a relational database

A database structure gives the data held within it context, producing information

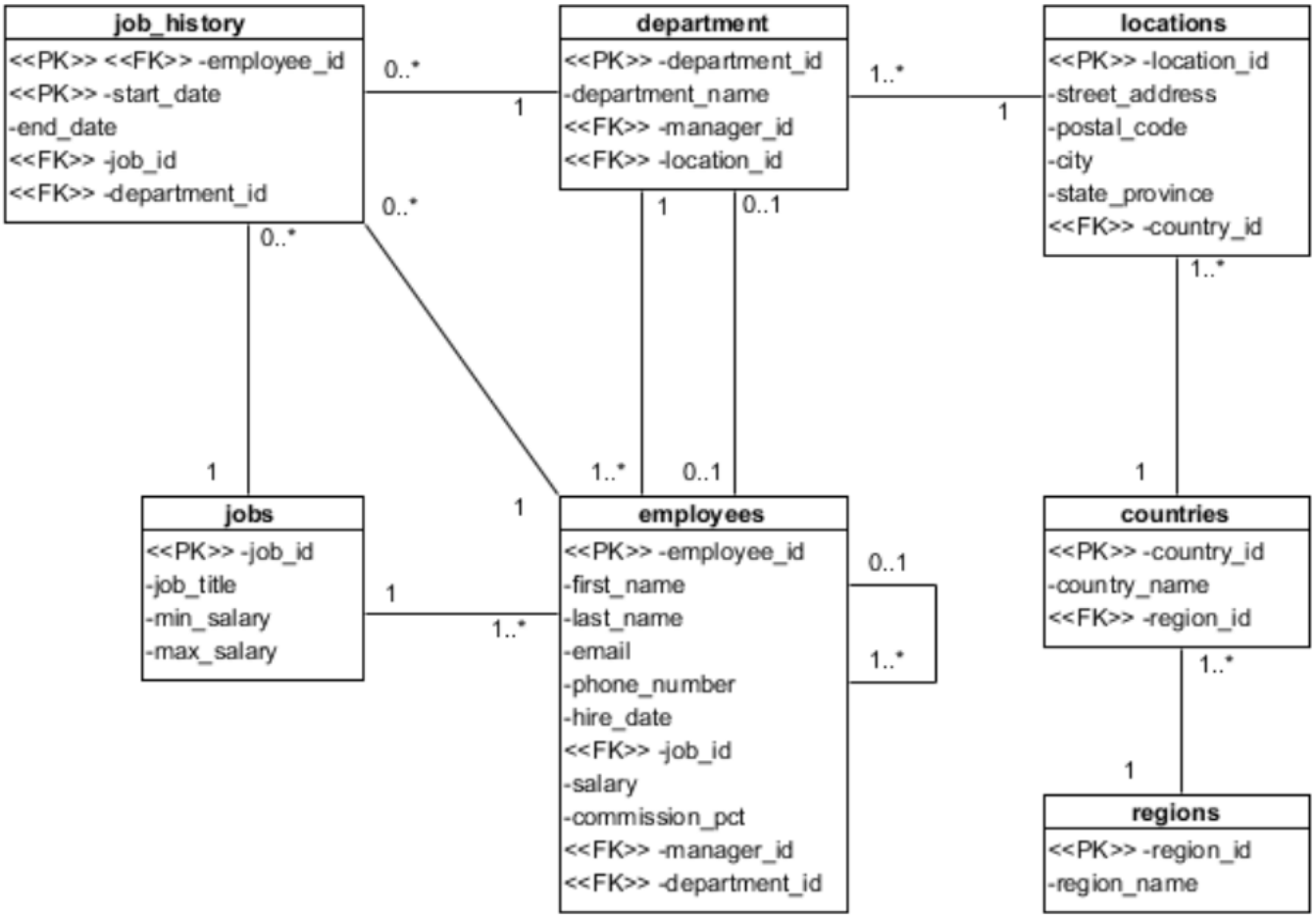
Database Management System (DBMS) is software that enables users to define, create, maintain and access a database



SQL

Before we start…

Many of the examples that we will use will come from the HR Schema database provided by Oracle



SQL Is:

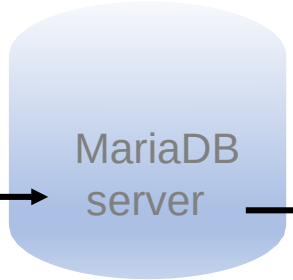
- Structured Query Language
- The standard for relational database management systems (RDBMS)
- Used in Oracle, MySQL, DB2, PostgreSQL, Microsoft SQL Server, Microsoft Access, Dbase, Paradox and many others
- Standard for information interchange as there is a version that runs on almost any platform and is recognised by almost any other Database.
- Often used as an interface between programs (written in Cobol, Java etc.) and databases. Also, between websites and databases
- First version 1970 by IBM named SEQUEL

What is SQL?

Structured Query Language (SQL) is a declarative language used to access data in an DBMS

Provides an interface to a relational database

```
SELECT department_name
FROM   departments;
```



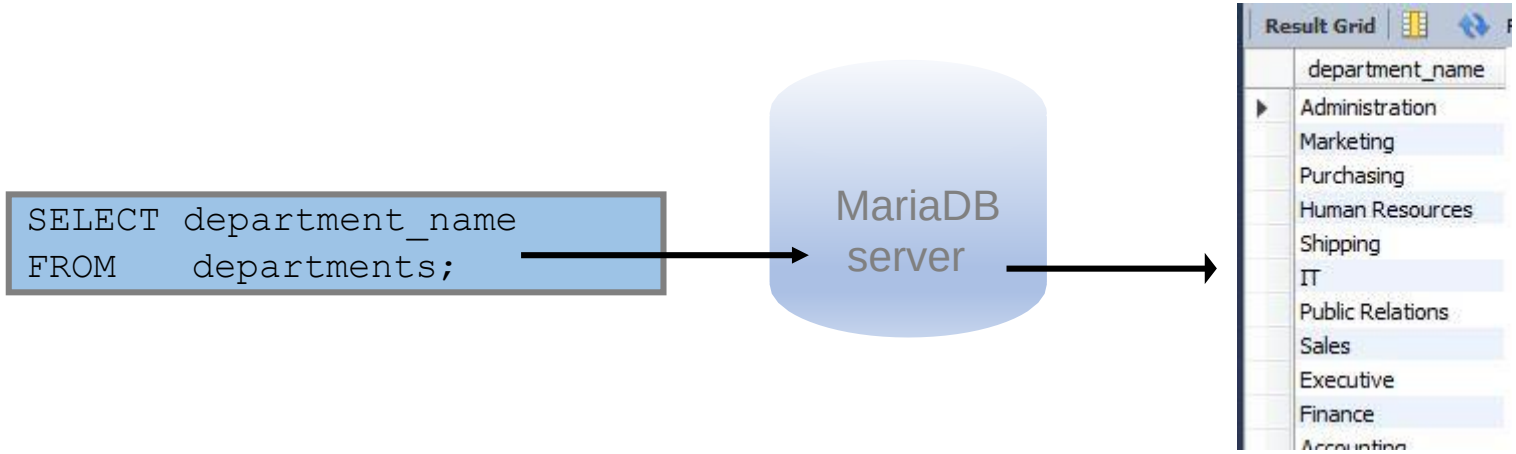
Result Grid

department_name
Administration
Marketing
Purchasing
Human Resources
Shipping
IT
Public Relations
Sales
Executive
Finance
Accounting

What is SQL?

Provides statements to help us work with the database

- Retrieving data stored in database (querying)
- Inserting, updating, deleting rows in a table
- Creating, altering, deleting tables and other objects
- Guaranteeing database consistency and integrity



SQL statement categories

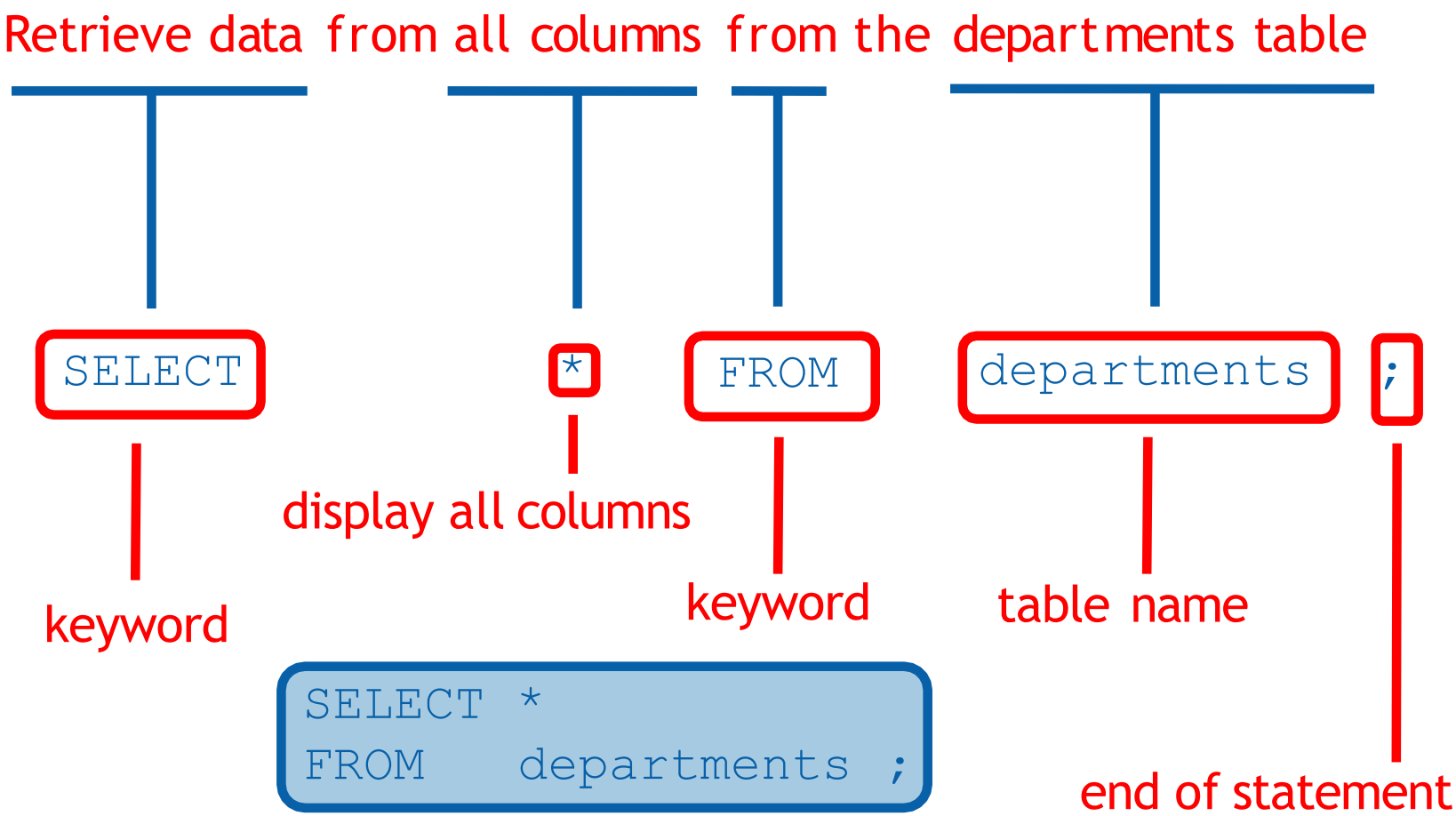
SELECT INSERT UPDATE DELETE REPLACE	Data manipulation language (DML)
CREATE ALTER DROP RENAME TABLE TRUNCATE TABLE	Data definition language (DDL)
GRANT REVOKE	Data control language (DCL)
COMMIT ROLLBACK SAVEPOINT	Transaction control

This is a database table for departments

DEPARTMENTS table

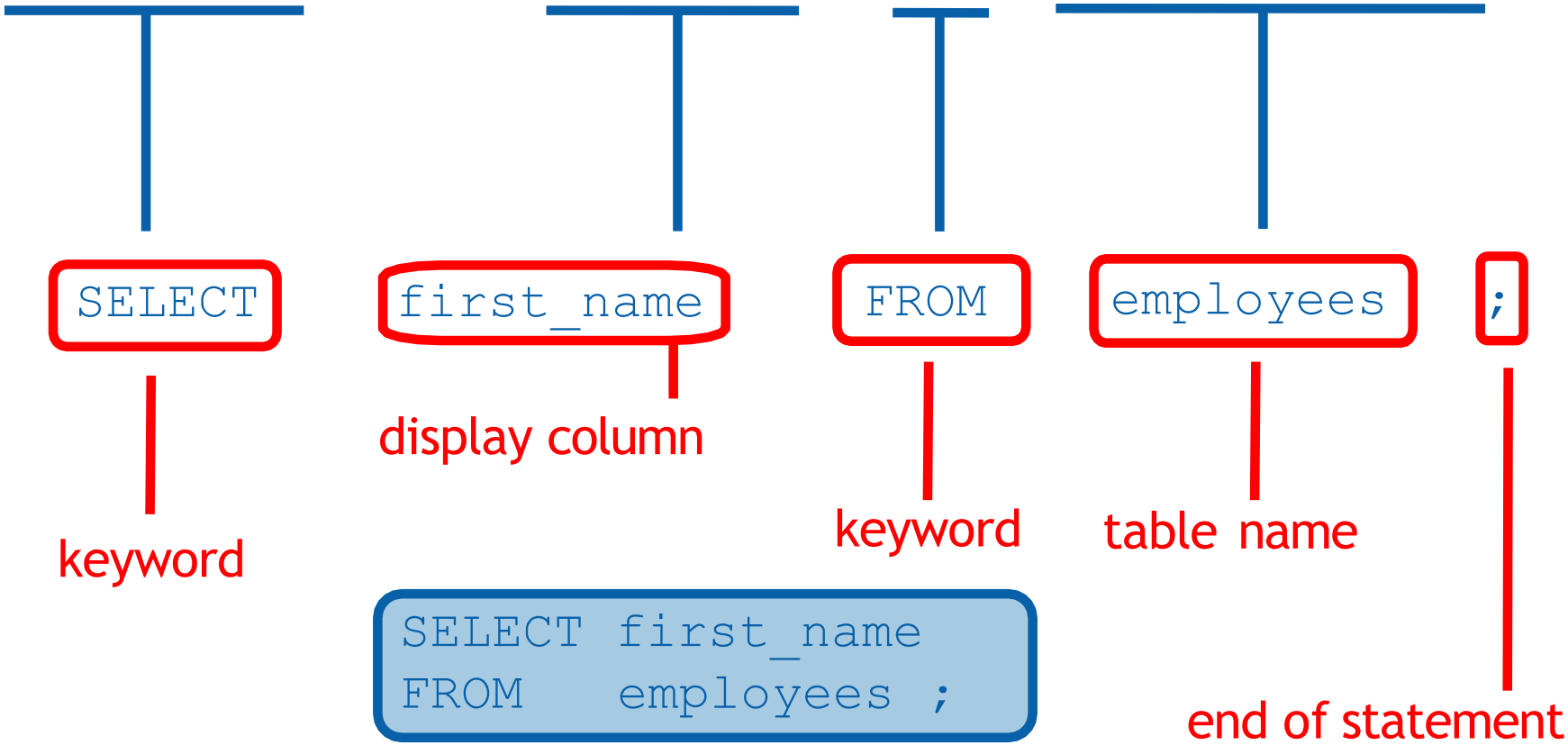
DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

SELECT statement structure



SELECT statement structure

Retrieve data from first name column from the employees table




```
SELECT first_name
FROM   employees ;
```

SELECT statement structure

Selection follows a standard command structure:

SELECT *attribute 1, attribute 2, attribute n, FUNC(attribute 3)*

FROM *tablename*

WHERE *condition*

ORDER BY *attribute 1, attribute 2, attribute n;*

Optionally **JOIN** commands may be used to join multiple tables together and the **GROUP BY** command can be used when aggregate functions are used (these will be covered later)

COMPARISON OPERATORS

Using Comparison Operators

EMPLOYEES

	EMPLOYEE_ID	FIRST_...	LAS...	EMAIL	PHONE_N...	HIRE_DATE	JOB_ID	SALARY	COMMISSIO...	MANAG...	DEPAR...
1	100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	(null)	(null)	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	(null)	100	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	(null)	100	90
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	(null)	102	60
5	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	(null)	103	60
6	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	(null)	103	60
7	124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	5800	(null)	100	50
8	141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	3500	(null)	124	50
9	142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	3100	(null)	124	50
10	143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	2600	(null)	124	50
11	144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	ST_CLERK	2500	(null)	124	50
12	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344....	29-JAN-00	SA_MAN	10500	0.2	100	80
13	174	Ellen	Abel	EABEL	011.44.1644....	11-MAY-96	SA_REP	11000	0.3	149	80
14	176	Jonathon	Taylor	JTAYLOR	011.44.1644....	24-MAR-98	SA_REP	8600	0.2	149	80
15	178	Kimberely	Grant	KGRANT	011.44.1644....	24-MAY-99	SA_REP	7000	0.15	149	(null)
16	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	(null)	101	10
17	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000	(null)	100	20
18	202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	6000	(null)	201	20
19	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000	(null)	101	110
20	206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300	(null)	205	110

Limiting Columns (Projection)

EMPLOYEES

Result Grid | Filter Rows: | Edit: | Export/Import:

employee_id	first_name	last_name	email	phone_number	hire_date	jo
100	Steven	King	SKING	515.123.4567	1987-06-17	AD
101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD
102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD

. . .

"retrieve some columns from Employees table"

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees;
```

Result Grid | Filter Rows: | Edit:

employee_id	last_name	job_id	department_id
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
105	Austin	IT_PROG	60

Limiting Rows (Selection): WHERE clause

```
SELECT column_name(s)
FROM   table_name
WHERE condition;
```

...the `WHERE` clause follows the `FROM` clause.

...restricts rows returned.

Limiting Rows (Selection)

EMPLOYEES

Result Grid | Filter Rows: | Edit:

employee_id	last_name	job_id	department_id
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
105	Austin	IT_PROG	60

"retrieve all employees in department 90"

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees
WHERE  department_id = 90;
```

Result Grid | Filter Rows: | Edit:

employee_id	last_name	job_id	department_id
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Comparison Operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
`BETWEEN ...AND...`	Between two values (inclusive)
`IN(set)`	Match any of a list of values
`LIKE`	Match a character pattern
`IS NULL`	Is a null value

Using Comparison Operators

EMPLOYEES

salary less or equal to 3000

	EMPLOYEE_ID	FIRST_...	LAS...	EMAIL	PHONE_N...	HIRE_DATE	JOB_ID	SALARY	COMMISSIO...	MANAG...	DEPAR...
1	100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	(null)	(null)	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	(null)	100	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	(null)	100	90
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	(null)	102	60
5	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	(null)	103	60
6	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	(null)	103	60
7	124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	5800	(null)	100	50
8	141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	3500	(null)	124	50
9	142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	3100	(null)	124	50
10	143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	2600	(null)	124	50
11	144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	ST_CLERK	2500	(null)	124	50
12	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344....	29-JAN-00	SA_MAN	10500	0.2	100	80
13	174	Ellen	Abel	EABEL	011.44.1644....	11-MAY-96	SA_REP	11000	0.3	149	80
14	176	Jonathon	Taylor	JTAYLOR	011.44.1644....	24-MAR-98	SA_REP	8600	0.2	149	80
15	178	Kimberely	Grant	KGRANT	011.44.1644....	24-MAY-99	SA_REP	7000	0.15	149	(null)
16	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	(null)	101	10
17	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000	(null)	100	20
18	202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	6000	(null)	201	20
19	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000	(null)	101	110
20	206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300	(null)	205	110

Using Comparison Operators

```
SELECT last_name, salary
FROM   employees
WHERE  salary <= 3000;
```

Result Grid | Filter R

last_name	salary
Baida	2900.00
Tobias	2800.00
Himuro	2600.00
Colmenares	2500.00
Mikkilineni	2700.00
Landry	2400.00

Membership Condition – IN operator

```
SELECT employee_id, last_name, salary, manager_id
FROM   employees
WHERE  manager_id IN (100, 101, 201);
```

Result Grid | Filter Rows: | Ed

employee_id	last_name	salary	manager_id
101	Kochhar	17000.00	100
102	De Haan	17000.00	100
108	Greenberg	12000.00	101
114	Raphaely	11000.00	100
120	Weiss	8000.00	100
121	Fripp	8200.00	100
122	Kaufling	7900.00	100
123	Vollman	6500.00	100

Range Condition – BETWEEN operator

```
SELECT last_name, salary
FROM   employees
WHERE  salary BETWEEN 2500 AND 3500;
```

lower limit upper limit

Result Grid | Filter Ro

last_name	salary
Khoo	3100.00
Baida	2900.00
Tobias	2800.00
Himuro	2600.00
Colmenares	2500.00
Nayer	3200.00
Mikkilineni	2700.00

Note: Between is inclusive of border values

Pattern matching – LIKE operator with wildcards

```
SELECT ProductName, Price
FROM   products
WHERE  ProductName LIKE '%tofu%';
```

Can have any character(s) before "tofu"

Can have any character(s) after "tofu"

Result Grid | Filter

product_name	price
Tofu	23.25
Longlife Tofu	10.00

Strings in comparisons

```
SELECT  ProductName,  Price
FROM    products
WHERE   ProductName = BINARY 'Tofu' ;
```

To force a case sensitive comparison for “LIKE” or “=“ add “BINARY” cast operator

Pattern matching – LIKE operator with wildcards

```
SELECT ProductName, Price  FROM products
WHERE  ProductName LIKE '_o%';
```

show all products with names:

beginning with any letter/number ("_")

followed by an "o"

and having any letter(s) or number(s) after that ("%")

Note: for 3rd letter to be 'o' , write as ' _ _o% ' with no spaces between the '_'

Pattern matching – LIKE operator

```
SELECT ProductName, Price  FROM products
WHERE  ProductName LIKE '_o%';
```

Result Grid | Filter Rows:

product_name	price
Tofu	23.25
Longlife Tofu	10.00
Northwoods Cranberry Sauce	40.00
Konbu	6.00
Nord-Ost Matjeshering	25.89
Gorgonzola Telino	12.50
Boston Crab Meat	18.40

Using Arithmetic Expressions

```
SELECT last_name AS “Last Name”, (salary * 1.05) AS "Salary
with 5% Raise"
FROM      employees
WHERE     job_id = SA_REP;
```

Alias for more readable output

Result Grid | Filter Rows:

Last Name	Salary with 5% Raise
Tucker	10500.0000
Bernstein	9975.0000
Hall	9450.0000
Olsen	8400.0000
Cambrault	7875.0000
Tuvault	7350.0000
King	10500.0000

We want to find the salary of all sales representatives if they are given a 5% raise

Using Arithmetic Expressions

```
SELECT last_name, ROUND((salary * 1.05),2) AS "Salary with
5% Raise"
FROM      employees
WHERE     job_id = SA_REP;
```

Sets the number of decimal points

Result Grid | Filter Rows:

last_name	Salary with 5% Raise
Tucker	10500.00
Bernstein	9975.00
Hall	9450.00
Olsen	8400.00
Cambrault	7875.00
Tuvault	7350.00
King	10500.00

Specify number of decimal points by rounding. ROUND is a single row function.

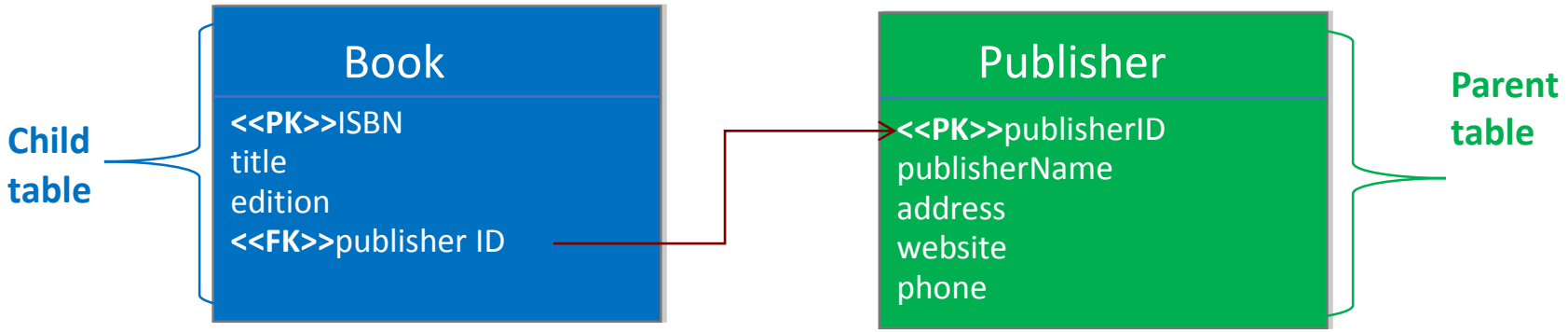
INTEGRITY

Entity Integrity

- Primary keys (PKs) cannot be null and must have a proper value

Referential Integrity (RI)

- **Each foreign key value** in a (child) table **exists as a primary key** in the referenced (parent) table
- So, when adding a new record to the child table (Book), if a foreign key value is entered, it must exist in the related primary key field of the parent table (Publisher)
- Referential Integrity maintains consistency among records in different tables



Referential Integrity Broken

203 (Pearson Education) has been deleted

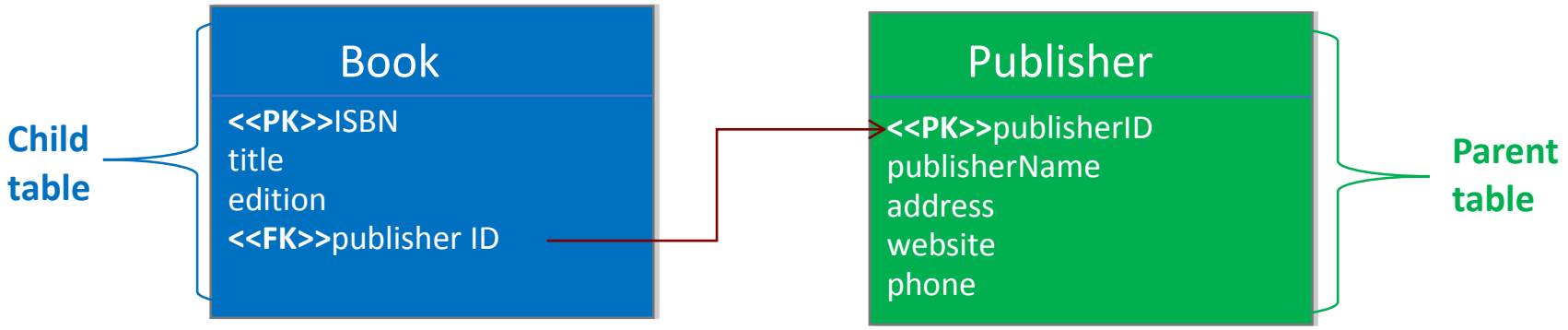
Link broken!

publisherID	publisherName	...
201	Palgrave MacMillan	
205	Prentice Hall	
206	Addison Wesley	

ISBN	title	edition	publisherID	...
9780230370500	An Introduction to Information Systems	2	201	
9780160840690	Systems Development Secrets	1	206	
9780221317801	Brief Guide to UML	3	201	
9780244708291	Database Theory	1	203	

Referential Integrity

- Foreign key fields **can** contain **NULL** values
- Primary key field values **can never** contain NULL values as they're needed to uniquely identify records
- The term **"cascade"** implies that changes to data in parent tables are propagated to all child tables



ARCHITECTURES

ANSI/SPARC 3-Level Architecture

Cobol language. Can be done in SQL

EXTERNAL LEVEL (users' views)

How end users see it

CONCEPTUAL LEVEL (community view)

How SQL users see it

INTERNAL LEVEL (storage view)

How computer system stores it

VIEW 1 | **VIEW 2**

Structural Properties & Constraints

Logical Representation of Stored Records

VIEW 1

```
01 BOOK
02 TITLE PIC X(30)
02 AUTHOR_NAME
PIC X(20)
```

VIEW 2

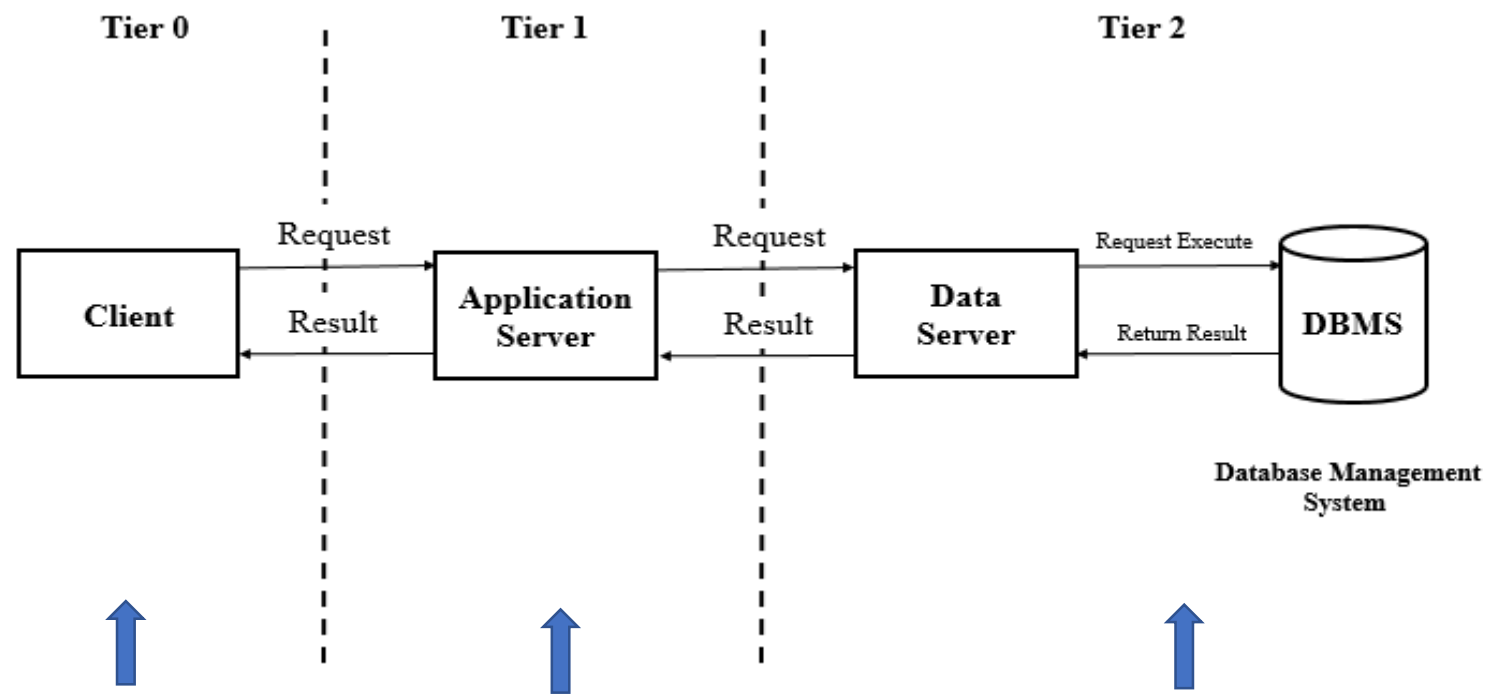
```
01 BOOK
02 TITLE PIC X(30)
02 PUBLISH_NAME
PIC X(20)
02 YEAR PIC 9(4)
```

```
BOOK
TITLE VARCHAR2(30)  AUTHOR
VARCHAR2 (20)  PUBLISHER
VARCHAR2(20)  YEAR NUMBER
(4)
  BOOKSHOP
••••••••••••••••••••••••••••••
```

```
STORED_BOOK     LENGTH=78  PREFIX
                TYPE=BYTE(6)
TITLE           TYPE=BYTE(30), INDEX=TITLE X
AUTHOR      TYPE=BYTE(20)
etc ••••••••••••••••••••••••••••
```

Context: 3(or n)-tier System Architecture

e.g., XAMPP stack



Presentation Tier

e.g., Webpage, GUI

Application Tier

e.g., Apache server running PHP

Data Tier

e.g., Oracle server

Separation of Components

If front and back end are separated, we can have multiple interfaces to the same server.

Oracle	MySQL	MS SQL Server
SQL Developer	MySQL Workbench	MS SQL Server Management Studio
Oracle Command Line	MySQL Command Line	SQL Command Line (sqlcmd)
Oracle Application Express (APEX)	PHPMyAdmin	Limited - MyLittleAdmin

Plus any custom interfaces we can develop.

Pros & Cons of 3-tier architecture

- Clients can be extremely thin (i.e. just a web browser – smartphones, tablets etc.)
- Load is spread between servers. Business logic and processing handled by the application server
- Scalable and flexible due to the system's modular design (load balancing, failovers etc.)
- We can break down the system to the nth degree (providing the software can handle it)
- Enabling a web interface immediately raises security issues

DBMS
PROS & CONS

Advantages of a DBMS

- Controlled redundancy (centralized, single data source)
- Consistency (e.g. integrity constraints, referential constraints)
- Accessible data can be shared across many applications
- Enforcement of standards (because it is centralised)
- Security

Disadvantages of a DBMS

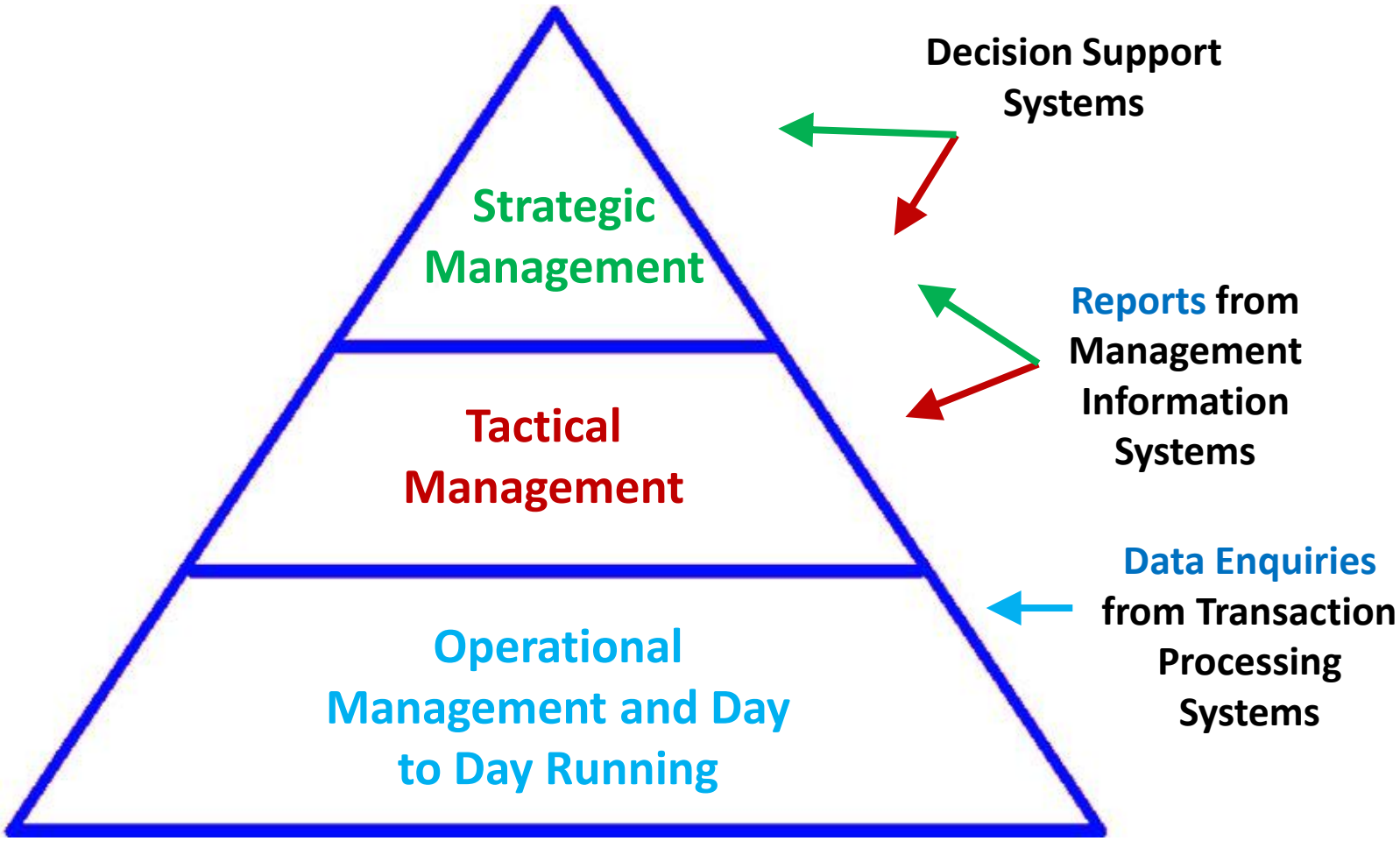
- Size (complex software)
- Complexity (requires expertise to use it)
- Additional hardware requirements (lots of storage space needed, separate server?)
- Recovery more difficult than std file system
- Higher impact of failure (because it is centralized)

CONTEXT: MANAGEMENT INFORMATION

Context

- Databases queries are used by all sorts of programs
- These applications can have all sorts of purposes
- But… one major area where they are essential is in running and managing organisations

Information and Systems at Different Levels of Organisation



Operational Information: Data Enquiries

Examples:

- a member of shop staff might need to know whether a particular item of stock is available
  - e.g. product enquiry by product name/code
- a librarian might need to find out whether a book is coming back soon
  - e.g. reservation enquiry by book title/code
- an engineer (working for a manufacturer) might need to find out whether a particular part is in stock
  - e.g. part enquiry by part number

Mgt. Information Systems

- Facilitate routine summarising and reporting
- Analysis, Exception, Key Target
- Assist tactical planning
- For short to medium term forecasting and budgeting

Strategic and Tactical Info:
Analysis Reports

Sales Analysis by Product Category 03 Apr 11
SE England

	January	February	March
Fresh Meat	24,000	23,000	9,000
Fruit & Veg	15,000	17,000	16,000
Frozen Goods	8,000	10,000	12,000
Dry Goods	31,000	33,000	35,000
Other Goods	19,000	20,000	22,000
Total	97,000	103,000	94,000

Strategic and Tactical Info:
Exception Reports

Highlight unusual cases.

```
Unpaid Invoice Report                    03 Apr 11
(Invoices unpaid after 60 days)

Customer  164923  Mighty Meat Ltd    *Order Stopped*

  Inv No.    Date        Total          No.     Paid
  6023465    15.05.10    7,026.00        4      Y
  6133492    16.06.10   13,974.00        3      N
  6246555    14,07.10   18,127.00        2      N
  6319845    15.08.10   11,849.00        1      N

Customer  170029  Fishy Fish

  Inv No.    Date        Total          No.     Paid
  6137426    19.06.10    7,623.00        3      N
```

Strategic and Tactical Info:
Key Target Reports

Provide a comparison against performance indicators.

```
Key Target Report              03 Apr 11
SW Warehouse

                            Target   Actual

  Stock Availability:       98.00%   99.25%
  Orders Proc. day 1:       98.00%   83.96%
  Backorder Time:           2 days   3 days
  Write Offs:               2.00%    3.46%
```

Strategic and Tactical Info:
Ad-hoc Reports

Are produced for one off queries.

```
SELECT ordNo, ordDate, prodCode, prodDesc
FROM    ORDER, ORDERLINE, PRODUCT
WHERE  ORDERLINE(status) = 'B'
AND ORDER(ordDate) < 11JAN18
```

Decision Support Systems

- Assist tactical planning, as well as operational and strategic planning
- Not concerned with standard reports
- Interactive support:
    - Explore ’what if?’ scenarios
    - Allow goal seeking
    - Explore optimisation scenarios

Summary

- To implement and manage a database, use a DBMS
- SQL is used to add, update, delete and access data in a DBMS
- SELECT statements are used to retrieve information from a database
- WHERE statement restricts the rows returned in a SELECT
- Maintaining referential integrity means ensuring all references (via FKs) in the database are complete
- Databases make use of ANSI/SPARC 3-level architecture
- Databases are often used within the context of 3-tier architecture
- SELECT queries are often used in organizational management