(Free) Sample 5

Taken from:

E-book 5: SQL

Knowledge River Ltd

Welcome to **Knowledge River** – the on-line e-learning store for Computer Science practitioners, academics and students.

Thank you for taking the time to download and examine this free sample – based on one of our popular e-learning products. In this free sample we have tried to give you a flavour of the full document upon which it is based. If you like what you see and would like to purchase the full document then please follow the instructions on our website: **www.knowledge-river.co.uk**

Whilst every effort has been made to ensure the accuracy of this material and any included software code has been tested carefully, they are only intended for instructional and illustrative purposes and are not guaranteed for any particular application, purpose or function. Knowledge River Ltd does not offer any warranties or representations, nor do we accept any liabilities with respect to any code examples included here.

Please note that Knowledge River provides Computer Science educational material in electronic format – we do not offer an on-line IT consultancy or help-desk function and so we cannot get into detailed correspondence on specific technical issues. However, we would appreciate general feedback and comments on what you think of the material, its layout, its usefulness and how it could be improved etc.

Please email us at: **feedback@knowledge-river.co.uk**

Knowledge River Ltd - Usage Policy (How to use our products)

We hope you enjoy using these e-learning products and get a lot of benefit by doing so. However, these electronic resources do represent a very considerable investment in intellectual effort, time and money on our behalf and naturally we want to protect our intellectual property. These products are meant for *your own personal educational use* and should not be modified, altered, edited, re-sold or transferred to a third party.

Please use and enjoy our work but please respect our efforts too.

The table below summarizes what you can and cannot do with our e-learning products:

Action	Allowed?	Notes
Reading	Yes	Obviously! On-screen or off-screen (see printing).
Printing / Making a Hardcopy	Yes (Limited)	Only single copies for *your own personal use.* Multiple copies and/or photocopies intended for distribution or sale to third parties is NOT allowed.
Editing / Updating / Extending	No	The e-learning products you purchase from Knowledge River Ltd must NOT be edited or modified in any way. This is how they were intended to be and this is how we want them to stay. Edits are our responsibility.
Copying (Electronic)	Yes (Limited)	You may copy the purchased e-learning product for *your own personal use* within your own computing equipment and storage devices (eg on a desktop and laptop with a backup on a portable storage device). However, you must NOT make multiple copies of these products with the intention of passing these copies onto third parties (so doing copies for your friends or putting copies on a server for sale or distribution is not allowed). In short, keep one or two copies for *your own personal convenience* but do not give copies to other people. This rule applies regardless of whether or not you charge money for it.
Transferring / Passing On	No	Each e-learning product purchased is intended for *one individual* – so please do not pass copies onto third parties – let them purchase their own. This rule applies even if there is no money involved or financial gain for yourself – please do not pass these products around – ever. Knowledge River products are non-transferable.
Re-selling	No	It goes without saying that this is never allowed – taking a copy of our work and then selling it on is simply theft and is immoral - please don't even think about doing this.
Anything Else	No	Anything not already covered is not allowed – please treat our company and products with respect. Enjoy.

The full *E-book 5* contents are:

Like this sample, the full document is in PDF format and can be purchased individually at a modest price or you may like to buy several different documents and get *very substantial discounts* – full details on our website. Thank you once again for taking the time to examine this **Knowledge River** educational material – please have a look at our website or drop us an email.

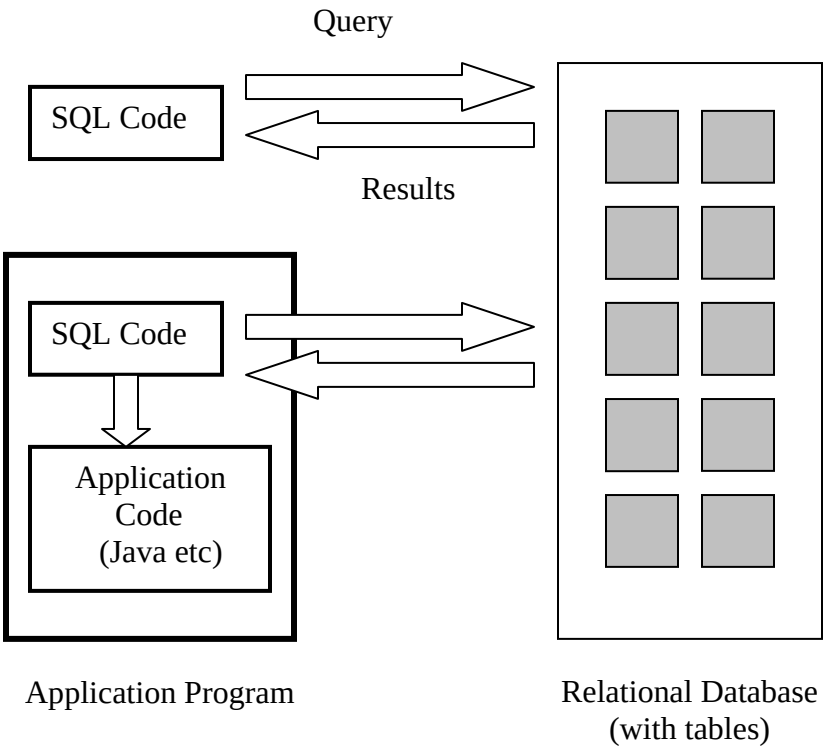
Below are three examples…

What does SQL stand for?

SQL stands for **Structured Query Language**. It is the industry-standard way of interfacing to relational databases. It is directly based on the relational theory we covered in e-book 3.

SQL can be used in its own right as a pure *query language* (to perhaps run a one-off ad-hoc query) or it may be embedded within another (host) language such as Java, C or a web-based environment to form a *database application* that is used on a regular basis. Either way, the database accepts the SQL code and returns a results set. If the SQL is running on its own, the results set will be displayed directly to the end-user. If the SQL is embedded within another (host) language then the results are passed across to the variables of the host language where they are further processed.

The diagram below shows how this works.



How do you create Tables?

Remember that the tables will be based directly on the normalized relations that we covered in the last e-book. Each relation that exists once the normalization process is complete will become a table.

When creating a brand new table here are the issues that you need to resolve (the following list assumes that you have the appropriate security privileges to create a new table and sufficient space to store the created & populated table):

- The name of the table (it must be unique within the schema)
- The number of columns in the table (as dictated by the normalized relations)
- The names of the columns (each column must be unique within the table)
- The data type of each column (using user-defined domains or built-in data types)
- The attributes making up the primary key
- The attributes making up any foreign keys
- Any other constraints (range checks, NULL, NOT NULL, etc)
- Any default values to be added

It is considered good practice to use your own user-defined domains if possible – so this must be done before starting to build the tables –as we have just demonstrated.

Here we implement the tables for the road haulage company. Note – we have used a mixture of built-in data types and our own user-defined domains. We have added some constraints using the CHECK key word (which can either be used on its own – as in the Driver table below – or as part of a separate named constraint – as in the Vehicle table and the later tables below). For ease of reference, all constraints and extra features such as defaults are in bold.

```
CREATE TABLE Driver
(Driver_ID          INTEGER          NOT NULL,
 Name               Staff_Name       NOT NULL,
 Address            VARCHAR(60)      NOT NULL,
 Telephone          INTEGER          NOT NULL,
 Licence_Renewed    DATE        NOT NULL    DEFAULT '01-JAN-2006',
 PRIMARY KEY (Driver_ID),
 CHECK (Driver_ID > 1))
```



```
CREATE TABLE Vehicle
(Registration       Vehicle_Registration      NOT NULL,
 Type               Vehicle_Type              NOT NULL,
 Max_Load           Payload                   NOT NULL,
 Service_Date       Service_Due,
 Fleet_Number       INTEGER                   NOT NULL,
 PRIMARY KEY (Registration),
 CONSTRAINT IsAValidVehicle
        CHECK Fleet_Number IN (1, 2, 3, 4, 5, 6, 7, 8))
```

```
CREATE TABLE Client
(Name               Client_Name
 Location           VARCHAR(50)      NOT NULL,
 Frequency          Pickups          NOT NULL   DEFAULT 'Daily',
 Service_type       Service          NOT NULL   DEFAULT 'Single Drop',
 Product            VARCHAR(30)      NOT NULL,
 PRIMARY KEY (Name))
```

```
CREATE TABLE Price_Bandings
(Band                   Price_Band          NOT NULL,
 Price_Per_Km           DECIMAL(4,2)        NOT NULL,
 PRIMARY KEY (Band),
 CONSTRAINT RightRate
        CHECK Price_Per_Km IN (0.35, 0.45, 0.65, 0.90, 1.10, 1.30))
```

How do we search with NULL values?

Remember – NULL is not actually a value. NULL means 'not known or not applicable'.

Consequently, a NULL will never be equal to a number, a date, a character string or anything else that has a real, concrete value. It is 'without value' and so cannot be directly compared against any other value. This conceptual point is very important when it comes to dealing with nulls in our database searches as we have to re-think how we construct our WHERE clauses.

For example, the following queries are **invalid**:

```
SELECT Registration, Fleet_Number
FROM Vehicle
WHERE Service_Date = NULL
```

```
SELECT Registration, Fleet_Number
FROM Vehicle
WHERE Service_Date <> NULL
```

Instead, SQL has two special predicates:

- IS NULL
- IS NOT NULL

Here is how to use it (re-writing the two previous queries):

```
SELECT Registration, Fleet_Number
FROM Vehicle
WHERE Service_Date IS NULL
```

```
SELECT Registration, Fleet_Number
FROM Vehicle
WHERE Service_Date IS NOT NULL
```

