

Higher National Unit Specification

General information for centres

Unit title: Programming in SQL

Unit code: DG03 34

Unit purpose: This Unit is designed to introduce candidates to programming in SQL. It is intended for candidates undertaking an HNC or HND in Computing, Computer Networking or a related area who require an understanding of SQL. On completion of the Unit candidates should be able to:

1. Describe the uses of database systems and SQL.
2. Create and use SQL queries.
3. Create and alter tables.
4. Insert, delete and update rows.
5. Use subqueries to retrieve and modify data.
6. Create views, indexes and transaction controls.
7. Join tables.

Credit value: 1 HN credit at SCQF level 7: (8 SCQF credit points at SCQF level 7)

**SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from Access 1 to Doctorates.*

Recommended prior knowledge and skills: Access to this Unit will be at the discretion of the Centre. There are no specific requirements but candidates would benefit from a basic knowledge of PC hardware and software. This may be demonstrated by the possession of HN Units such as DG0K 33 Hardware Concepts and DF9L 33 Operating System Concepts.

Core skills: There may be opportunities to gather evidence towards core skills in this Unit, although there is no automatic certification of core skills or core skills components.

Context for delivery: This Unit is included in the framework of a number of HNC and HND group awards. It is recommended that it should be taught and assessed within the context of the particular group award to which it contributes.

Assessment: Evidence for the knowledge and/or skills for the entire Unit must be produced using a set of 30 restricted-response questions to assess candidates' knowledge and understanding. This may be administered as a single end-of unit test, or as several subtests, each covering one or more outcomes.

General information for centres (cont)

Candidates must answer at least 70% of the questions correctly in order to obtain a pass. If subtests are used, they must also score at least 70% in each subtest.

Testing must take place in a closed-book environment where candidates have no access to books, handouts, notes or other learning material. Testing can be done in either a machine-based or paper-based format and must be invigilated by a tutor or mentor. There must be no communication between candidates and communication with the administrator must be restricted to matters relating to the administration of the test.

If a candidate requires to be reassessed, a different selection of questions must be used. At least half the questions in the reassessment must be different from those used in the original test.

If an outcome has a practical component, this must be assessed by having the candidate produce code fragments, illustrating particular SQL features. These code fragments may be combined to create a meaningful program, but this is not mandatory.

Higher National Unit specification: statement of standards

Unit title: Programming in SQL

Unit code: DG03 34

The sections of the Unit stating the Outcomes, knowledge and/or skills, and evidence requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the knowledge and/or skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Outcome 1

Describe the uses of database systems and SQL.

Knowledge and/or skills

- ◆ Define the term Relational Database
- ◆ Describe the use of databases in a business environment
- ◆ Describe the features of SQL

Evidence requirements

The knowledge and skills component of Outcome 1 must be examined by six questions, two derived from each of the three items listed below. Each question must be derived from a single item.

1. Define the term Relational Database

Codd model, data structures, storage and retrieval operations, integrity constraints

2. Describe the use of databases in a business environment

E-commerce / web-enabled applications, finance, human resources, project management, service automation, performance management, customer relationship management, supply chain management

3. Describe the features of SQL

Tables, columns, data types, rows, keys, indexes, views, SQL statements, transactions, relationships (1:1, 1:m, m:n)

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

Alternatively, the 6 questions for this outcome may contribute towards a single end-of-unit test of 30 questions.

Code fragments

The restricted-response questions should be supplemented by a number of code fragments, illustrating particular SQL features. In the case of this outcome, no code fragments are required.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 70 minutes should be allocated for a 30-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 2

Create and use SQL queries.

Knowledge and/or skills

- ◆ Use SELECT statement
- ◆ Use WHERE, ORDER BY, GROUP BY, and HAVING clauses
- ◆ Use comparison operators
- ◆ Create complex queries
- ◆ Use wildcards
- ◆ Use arithmetic operators in queries
- ◆ Format output

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 2 must be examined by seven questions, derived from the seven items listed below. Each question must be derived from a different item.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

1. Use SELECT statement

With single and multiple tables

2. Use WHERE, ORDER BY, GROUP BY, and HAVING clauses

3. Use Comparison Operators

=, <>, >, <, >=, <=

4. Create Complex Queries

Using NULL, NOT NULL, NOT, AND, OR, LIKE, IN, BETWEEN, functions (COUNT, SUM, AVG, MAX, MIN), date functions

5. Use wildcards

%, _ (underscore), dialect-dependent wildcards

6. Use arithmetic operators in queries

+, -, *, /

7. Format output from queries

String functions, mathematical functions

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 7 questions for this outcome may contribute towards a single end-of-unit test of 30 questions.

Code fragments

The restricted-response questions should be supplemented by a number of code fragments, illustrating particular SQL features. In the case of this outcome, at least one code fragment should be produced to illustrate each of the following features: SELECT statements, WHERE, ORDER BY, GROUP BY and HAVING clauses, comparison operators, complex queries, wildcards, arithmetic operators, formatted output.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 70 minutes should be allocated for a 30-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 3

Create and alter tables

Knowledge and/or skills

- ◆ Create a table
- ◆ Create keys
- ◆ Alter columns within a table
- ◆ Add rows to a table
- ◆ Drop a table

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 3 must be examined by five questions, derived from the five items listed below. Each question must be derived from a different item.

1. Create a table

CREATE TABLE statement, creating a table from another table

2. Create keys

UNIQUE, PRIMARY KEY constraints

3. Alter columns within a table

ALTER TABLE statement, add new columns, alter existing columns

4. Add rows to a table

INSERT INTO statement

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

5. Drop a table

DROP statement

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 5 questions for this outcome may contribute towards a single end-of-unit test of 30 questions.

Code fragments

The restricted-response questions should be supplemented by a number of code fragments, illustrating particular SQL features. In the case of this outcome, at least one code fragment should be produced to illustrate each of the following features: create a table, create keys, alter columns within a table, add rows to a table and drop a table.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 70 minutes should be allocated for a 30-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 4

Insert, delete and update rows.

Knowledge and/or skills

- ◆ Insert rows into a table
- ◆ Delete rows from a table
- ◆ Update rows within a table

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 4 must be examined by three questions, derived from the three items listed below. Each question must be derived from a different item.

1. Insert rows into a table

INSERT, INSERT ... SELECT statements

2. Delete rows from a table

DELETE statement, single and multiple rows

3. Update rows within a table

UPDATE statement, SET, WHERE clauses

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 3 questions for this outcome may contribute towards a single end-of-unit test of 30 questions.

Code fragments

The restricted-response questions should be supplemented by a number of code fragments, illustrating particular SQL features. In the case of this outcome, at least one code fragment should be produced to illustrate each of the following features: insert rows into a table, delete rows from a table, update rows within a table.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 70 minutes should be allocated for a 30-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

Outcome 5

Use subqueries to retrieve and modify data.

Knowledge and/or skills

- ◆ Describe the use of subqueries
- ◆ Use subqueries to break down and perform complex queries.

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 5 must be examined by two questions, one derived from each of the items listed below. Each question must be derived from a single item.

1. Describe the use of subqueries
2. Use subqueries to break down and perform complex queries.

Expression subqueries, qualified predicate subqueries, EXISTS and NOT EXISTS clauses, correlated and non-correlated subqueries.

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 2 questions for this outcome may contribute towards a single end-of-unit test of 30 questions.

Code fragments

The restricted-response questions should be supplemented by a number of code fragments, illustrating particular SQL features. In the case of this outcome, at least two code fragments should be produced to illustrate each of the following features: use subqueries to break down and perform complex queries.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 70 minutes should be allocated for a 30-question end-of-unit test.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 6

Create views, indexes and transaction controls

Knowledge and/or skills

- ◆ Create and drop indexes
- ◆ Create and drop views
- ◆ Lock tables
- ◆ Commit and rollback transactions

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 6 must be examined by four questions, derived from the four items listed below. Each question must be derived from a different item.

1. Create and drop indexes

CREATE INDEX, DROP INDEX statements

2. Create and drop views

CREATE VIEW, DROP VIEW statements

3. Lock tables

LOCK TABLE statement

4. Commit and rollback transactions

BEGIN TRANSACTION, COMMIT TRANSACTION, ROLLBACK TRANSACTION statements

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 4 questions for this outcome may contribute towards a single end-of-unit test of 30 questions.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

Code fragments

The restricted-response questions should be supplemented by a number of code fragments, illustrating particular SQL features. In the case of this outcome, at least one code fragment should be produced to illustrate each of the following features: create indexes, drop indexes, create views, drop views, lock tables, commit transactions, rollback transactions.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 70 minutes should be allocated for a 30-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

Outcome 7

Join tables.

Knowledge and/or skills

- ◆ Perform an inner join
- ◆ Perform an outer join
- ◆ Perform an equijoin

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 7 must be examined by three questions, derived from the three items listed below. Each question must be derived from a different item.

1. Perform an inner join
2. Perform an outer join
3. Perform an equijoin

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 3 questions for this outcome may contribute towards a single end-of-unit test of 30 questions.

Code fragments

The restricted-response questions should be supplemented by a number of code fragments, illustrating particular SQL features. In the case of this outcome, at least one code fragment should be produced to illustrate each of the following features: perform an outer join, perform an inner join, perform an equijoin.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 70 minutes should be allocated for a 30-question end-of-unit test.

Higher National Unit specification: statement of standards (cont)

Unit title: Programming in SQL

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Administrative Information

Unit code:	DG03 34
Unit title:	Programming in SQL
Superclass category:	CD
Date of publication:	May 2004
Version:	01
Source:	SQA

© Scottish Qualifications Authority 2004

This publication may be reproduced in whole or in part for educational purposes provided that no profit is derived from reproduction and that, if reproduced in part, the source is acknowledged.

Additional copies of this Unit specification can be purchased from the Scottish Qualifications Authority. The cost for each Unit specification is £2.50. (A handling charge of £1.95 will apply to all orders for priced items.)

Higher National Unit specification: support notes

Unit title: Programming in SQL

This part of the Unit specification is offered as guidance.

The support notes are not mandatory.

While the exact time allocated to this Unit is at the discretion of the centre, the notional design length is 40 hours.

The suggested time allocation for each outcome (including assessment) is as follows:

Outcome 1:	4 hours
Outcome 2:	10 hours
Outcome 3:	4 hours
Outcome 4:	4 hours
Outcome 5:	6 hours
Outcome 6:	6 hours
Outcome 7:	6 hours

Guidance on the content and context for this Unit

As it is likely that the bulk of the material in this Unit will be delivered through lecturer exposition, it is important that every opportunity is taken to introduce real-world examples, opportunities for whole-class and group discussion and practical demonstrations wherever possible. Concepts and terminology should be presented in context throughout the Unit. Video presentations should be used where appropriate for providing an alternative explanation of a difficult topic, or as a focus for class discussion or groupwork.

Given the theoretical nature of this Unit, it is intended that a significant amount of time will be made available as a central part of the course for revision, tutorials and formative assessment exercises. Candidates should be strongly encouraged to undertake further reading, and opportunities for individual or group research should be provided.

The most important overall emphasis should be on the relevance and currency of content in such a rapidly-evolving field.

Outcome 1

1 Define the term Relational Database

Candidates should be aware that a Relational Database is one based on the model developed by E.F. Codd, allowing the definition of data structures, storage and retrieval operations and integrity constraints. In such a database the data is organised in tables, which are collections of rows or records. Each row in a table contains the same fields. Certain fields may be designated as keys, meaning that searches for specific values of that field will use indexes to improve performance.

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

2 Describe the use of databases in a business environment

Candidates should be familiar with the use of e-commerce and web-enabled applications in areas such as finance, human resources, project management and planning, service automation, performance management, customer relationship management and supply chain management, for communications, consumer products, government, financial services, health care, technology, utilities and other industries.

3 Describe the features of SQL

Candidates should also be aware that the data stored in SQL databases is organised in tables. Each table consists of a number of columns with an associated data type. Each table can have a number of rows. Indexes can be created on certain columns to provide high speed access to rows. Views can be created to allow access to parts of a database. Databases can be accessed and manipulated by issuing SQL statements to retrieve, insert, update and delete data and to administer tables and indexes. Data retrieved from databases is returned as tables. SQL ensures the integrity and consistency of the database up to the end of the last committed transactions, by ensuring that all enclosed operations will all be executed or none of them. The SQL relational model allows representation of 1:1, 1:many and many:many relationships.

Outcome 2

1 Use SELECT statement

Candidates should be aware that the SELECT query is used to retrieve records from a database.

2 Use WHERE, ORDER BY, GROUP BY, and HAVING clauses

Candidates should be able to use the FROM, WHERE, GROUP BY, ORDER BY HAVING and LIMIT keywords.

3 Use comparison operators

Candidates should be able to use comparison operators to limit the records returned with the WHERE clause: equal to (=), not equal to (<> or !=), less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=).

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

4 Create complex queries

Candidates should be aware that the NULL value is always false in comparison to any other value and be able to create complex queries by using NULL and NOT NULL and the logical operators NOT, AND and OR. They should be able to use LIKE (to compare strings), BETWEEN (to check for values between a range, IN (to check for values within a list) and NOT IN (to check that a value is not in a list) and use the aggregate functions (COUNT, SUM, AVG, MAX, MIN) and date and time functions (DATE, DAY, DAYNAME, DAYOFWEEK, DAYOFMONTH, DAYOFYEAR, WEEK, MONTH, MONTHNAME, YEAR, TIME, HOUR, MINUTE, SECOND).

5 Use wildcards

Candidates should be able to use SQL wildcards, including:

- % any string of zero or more characters
- _ any single character
- [] Any single character within the specified range ([a-f]) or set ([abcdef])
- [^] Any single character not within the specified range ([^a-f]) or set ([^abcdef]).

Wildcards may vary depending on the dialect of SQL being used. The first two listed above are standard, the last two are from Microsoft Transact-SQL.

6 Use arithmetic operators in queries

Candidates should be able to use the standard arithmetic operators: add (+), subtract (-), multiply (*) and divide (/) and be aware of the order of precedence when these are used.

7 Format output

Candidates should also be able to format output from queries, using string and mathematical functions, eg: string functions: LEFT, RIGHT, SUBSTRING, CHAR etc., mathematical functions: ABS, RAND, ROUND, SQUARE etc.

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

Outcome 3

1 Create a table

Candidates should be able to create new tables using the CREATE TABLE statement, eg: to create an ORDERS table:

```
CREATE TABLE ORDERS
(CUSTOMER_ID INTEGER NOT NULL,
STOCK_CODE CHAR(10) NOT NULL);
```

They should know that the CREATE TABLE statement can be used to create a table from another table. The CREATE TABLE statement allows a query to be specified as the source of the columns and rows for the new table. If the query used returns rows, the rows are inserted into the new table.

2 Create keys

Candidates should also be able to declare unique or primary key constraints to ensure that no two rows in a table have the same values in the specified columns, eg:

```
CREATE TABLE ORDERS
(CUSTOMER_ID INTEGER NOT NULL,
STOCK_CODE CHAR(10) NOT NULL,
PRIMARY KEY (CUSTOMER_ID));
```

They should be aware that a table's primary key constraint is normally used in conjunction with referential integrity constraints defined on other tables.

3 Alter columns within a table

Candidates should also be able to use the ALTER TABLE statement to add new columns to a table, or alter existing columns, eg:

```
ALTER TABLE ORDERS ADD (UNIT_PRICE DECIMAL (8,2) NULL);
```

4 Add rows to a table

They should be able to use the INSERT INTO statement to add rows to a table, eg:

```
INSERT INTO ORDERS VALUES (124, 'DVD15', 14.95)
```

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

5 Drop a table

Candidates should also be able to use the DROP TABLE statement to remove a table definition and all associated data, indexes and constraints and be aware that any view or stored procedure that references the dropped table must be explicitly dropped by using the DROP VIEW or DROP PROCEDURE statement.

Outcome 4

1 Insert rows into a table

Candidates should be able to use the INSERT INTO statement to add rows to a table, eg:

```
INSERT INTO ORDERS VALUES (124, 'DVD15', 14.95)
```

They should be aware that a SELECT clause in the INSERT statement can be used to pull values into a table from one or more other tables.

The SELECT clause can insert values into some or all of the columns in a row. If the columns are in the same order in their CREATE TABLE statements, there is no need to specify column names in either table. If a table named NEW_ORDERS contains rows of order information in the same format as in ORDERS then to add all the rows in NEW_ORDERS to the ORDERS table:

```
INSERT ORDERS
SELECT *
FROM NEW_ORDERS
```

1 Delete rows from a table

Candidates should be able to use the DELETE statement to delete single and multiple rows from a table. A simplified version of the DELETE syntax is:

```
DELETE table_name
WHERE column_name = expression
```

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

1 Update rows within a table

Candidates should be able to use the UPDATE statement to change single rows, groups of rows, or all rows in a table. UPDATE specifies the row or rows to be changed and the new data, which can be a specified constant or expression or data pulled from other tables. A simplified version of the UPDATE syntax for updating specified rows is:

```
UPDATE table_name
  SET column_name = expression
  WHERE search_conditions
```

The SET clause specifies the columns to be amended and the changed values. The WHERE clause determines which row or rows are to be updated. If the WHERE clause is omitted, the specified columns of all the rows are updated with the values given in the SET clause.

Outcome 5

1 Describe the use of subqueries

Candidates should be aware that subqueries are queries that appear within a WHERE or HAVING clause of another SQL statement and that they can be used to handle query requests that are expressed as the results of other queries.

2 Use subqueries to break down and perform complex queries.

Candidates should know that a subquery can also be used almost anywhere an expression is allowed. SELECT statements that contain one or more subqueries are sometimes called “nested queries” or “nested select statements”. They should know that there are two basic types of subqueries:

- **expression subqueries** which are introduced with an unmodified comparison operator, and must return a single value;
- **quantified predicate subqueries** which operate on lists introduced with IN or with a comparison operator modified by ANY or ALL or subqueries introduced with EXISTS..

Subqueries of either type can be correlated (repeating) or non-correlated.

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

Outcome 6

Candidates should be able to create views, indexes and transaction controls.

1 Create and drop indexes

Candidates should know that an index provides quick access to data in a table, based on the values in specified columns. A table can have more than one index. Indexes speed up data retrieval by pointing to the location of a table column's data on disk. For example, suppose you need to run frequent queries using the CUSTOMER_ID of customers in the ORDERS table. To prevent the server from having to search through each row in the table, which could be time-consuming if the table has millions of rows, you could create an index, called CUSTOMER_ID_IND:

```
CREATE INDEX CUSTOMER_ID_IND
ON ORDERS (CUSTOMER_ID);
```

The index is used automatically the next time you query the CUSTOMER_ID column in ORDERS. SQL has no syntax for referring to an index in a query. You can only create or drop indexes from a table; the SQL server decides whether or not to use the indexes for each query submitted for that table. As the data in a table changes over time, the SQL server may change the table's indexes to reflect those changes. There are several different types of indexes:

- **Composite indexes** involve more than one column. They are used when two or more columns are best searched as a unit because of their logical relationship.
- **Unique indexes** do not allow any two rows in the specified columns to have the same value. The SQL Server checks for duplicate values when the index is created and each time data is added.
- **Clustered indexes** force the SQL Server to sort and re-sort the rows of a table so that their physical order is always the same as their logical (or indexed) order. There can only be one clustered index per table.
- **Non-clustered indexes** do not require the physical order of rows to be the same as their indexed order. Each non-clustered index can provide access to the data in a different sort order.

Indexes can be created by using the CREATE INDEX statement or by using the UNIQUE or PRIMARY KEY constraints of the CREATE TABLE statement.

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

2 Create and drop views

Candidates should be aware that a view is a named SELECT statement stored in a database as an object that allows a subset of rows or columns in one or more tables to be viewed. Views can be used to customise a user's perception of the tables in a particular database and improve security by allowing users access only to the data they require. They are created using the CREATE VIEW statement and dropped using the DROP VIEW statement.

3 Lock tables

Candidates should know that the LOCK TABLE statement allows a user to request that before a table is accessed, a lock is put on it for the duration of a transaction. This can be useful when an immediate table lock will reduce the overhead of acquiring a large number of row or page locks and save overall locking time. If a table lock is not explicitly requested, a scan acquires page or row locks until it reaches the table's lock promotion threshold, at which point it tries to acquire a table lock. The syntax of the LOCK TABLE statement is:

```
LOCK TABLE table_name IN {SHARE | EXCLUSIVE} MODE  
[WAIT [no_of_seconds] | NOWAIT]
```

The WAIT/NOWAIT option allows the user to specify how long the command waits to acquire a table lock if it is blocked.

4 Commit and rollback transactions

Candidates should be able to commit and rollback transactions. They should know that a transaction groups a set of SQL statements together, so that they are treated as a unit. Either all statements in the group are executed or none of them are executed. The BEGIN TRANSACTION and COMMIT TRANSACTION statements tell the SQL Server to process a number of individual commands as a single unit. ROLLBACK TRANSACTION undoes the transaction, either back to its beginning, or back to a savepoint, defined inside a transaction by using SAVE TRANSACTION. Transactions give the user control over transaction management. In addition to grouping SQL statements to behave as a single unit, they improve performance, since system overhead is incurred once per transaction, rather than once for each individual command.

Outcome 7

Candidates should be aware that where fields in two different tables take values from the same set, a JOIN operation can be performed to select related records in the two tables by matching values in those fields. The fields will often have the same name in both tables. For example, an ORDERS table might contain (CUSTOMER_ID, STOCK_CODE) and an INVENTORY table might contain (STOCK_CODE, UNIT_PRICE) so to calculate a given customer's bill you would sum the prices of all products ordered by that customer by joining on the STOCK-CODE fields of the two tables. This can be extended to joining multiple tables on multiple fields.

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

1 Perform an inner join

An **inner join** (sometimes called a "simple join") is a join of two or more tables that returns only those rows that satisfy the join condition.

2 Perform an outer join

An **outer join** extends the result of a simple join. An outer join returns all rows that satisfy the join condition and also returns some or all of those rows from one table for which no rows from the other satisfy the join condition.

Higher National Unit specification: support notes (cont)

3 Perform an equijoin

An **equijoin** is a join with a join condition containing an equality operator. An equijoin combines rows that have equivalent values for the specified columns.

Guidance on the delivery and assessment of this Unit

This Unit is likely to form part of a group award which is primarily designed to provide candidates with technical or professional knowledge and skills related to a specific occupational area. It is highly technical in content and should not be adopted by group awards in other areas or delivered as a stand-alone Unit without careful consideration of its appropriateness. It is a Unit which candidates are likely to find accessible at an introductory level; it is suggested that it be delivered as part of a first-year or second-year HNC/HND program in Computing or a related area. It should be delivered in tandem with other Computing Units and opportunities for teaching and assessment integration explored.

To minimise assessment overhead, sets of restricted-response questions are used to provide evidence of candidates' knowledge for all Outcomes. It is suggested that multiple-choice questions can be used as the preferred assessment method – as well as reducing the time required for assessment and marking, these reduce the need for candidates to memorise details and encourage understanding. The number of questions which must be answered correctly corresponds to 70% of those set in each case.

The restricted-response questions should be supplemented by collection of code fragments, illustrating particular SQL features. These code fragments may be combined to create a meaningful program, but this is not mandatory.

Open learning

If this Unit is delivered by open or distance learning methods, additional planning and resources may be required for candidate support, assessment and quality assurance.

Higher National Unit specification: support notes (cont)

Unit title: Programming in SQL

A combination of new and traditional authentication tools may have to be devised for assessment and re-assessment purposes.

For further information and advice, please see *Assessment and Quality Assurance for Open and Distance Learning* (SQA, February 2001 — publication code A1030).

Special needs

This Unit specification is intended to ensure that there are no artificial barriers to learning or assessment. Special needs of individual candidates should be taken into account when planning learning experiences, selecting assessment instruments or considering special alternative Outcomes for Units. For information on these, please refer to the SQA document *Guidance on Special Assessment Arrangements* (SQA, 2001).

General information for candidates

Unit title: Programming in SQL

This is a 1-credit Unit at Level 7 intended for candidates undertaking a Computing or IT-related qualification who require an understanding of Desktop Operating Systems. It is designed to develop an understanding of the issues involved in installing and administering a desktop operating system. On completion of the Unit you should be able to:

- Describe the uses of database systems and SQL.
- Create and use SQL queries.
- Create and alter tables.
- Insert, delete and update rows.
- Use subqueries to retrieve and modify data.
- Create views, indexes and transaction controls
- Join tables.

In the first part of the course, you will study the uses of database systems and SQL. This includes defining the term relational database, describing the use of databases in a business environment and describing the features of SQL.

The second section covers create and using SQL queries, including using the SELECT statement, using the WHERE, ORDER BY, GROUP BY, and HAVING clauses, using comparison operators, creating complex queries, using wildcards, using arithmetic operators in queries and formatting output.

The third section covers creating and altering tables, including creating a table, creating keys, altering columns within a table, adding rows to a table and dropping a table.

The fourth section covers inserting, deleting and updating rows, including inserting rows into a table, deleting rows from a table and updating rows within a table.

The fifth section covers using subqueries to retrieve and modify data, including describing the use of subqueries and using subqueries to break down and perform complex queries.

The sixth section covers create views, indexes and transaction controls, including creating and dropping indexes, creating and dropping views, locking tables and committing and rolling back transactions.

The final section covers joining tables, including performing an outer join, performing an inner join and performing an equijoin.

There will be a closed-book multiple-choice assessment covering all outcomes. You will be presented with 30 questions and expected to answer 70% of these correctly. You will also be expected to produce a collection of code fragments, illustrating particular SQL features.

You must satisfy the requirements for these assessments in order to achieve the Unit.