

8.1 Programming languages



This section has been set as optional by your instructor.

Programming languages

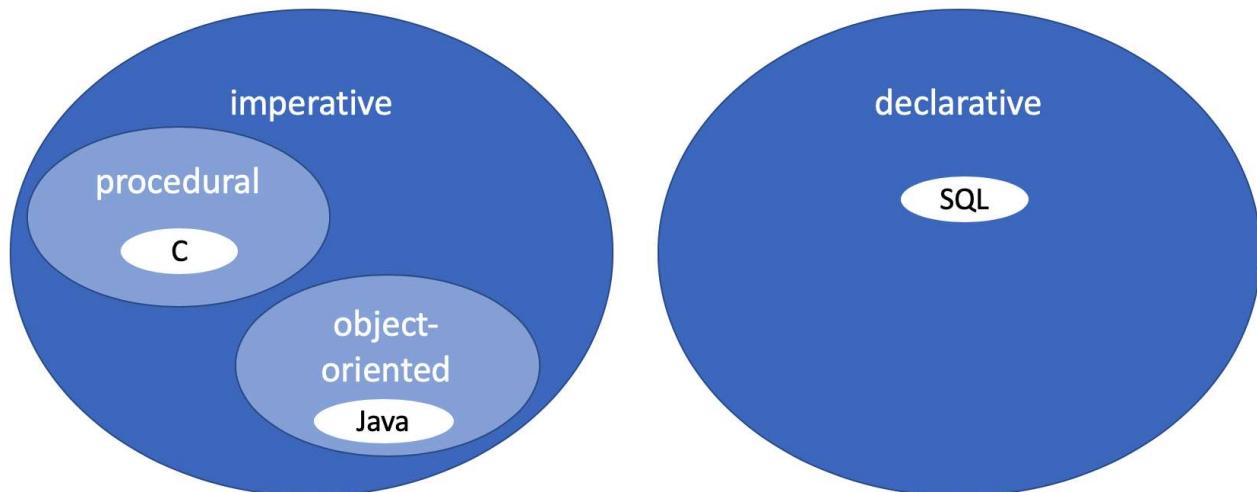
Programming languages fall into two broad categories, or paradigms: imperative and declarative.

Imperative languages contain control flow statements that determine the execution order of program steps. Control flow statements include loops for repeatedly executing code and conditionals for conditionally executing code. Two popular types of imperative languages are:

1. **Procedural languages** are composed of procedures, also called functions or subroutines. Most languages developed prior to 1990 are procedural. Ex: C and COBOL.
2. **Object-oriented languages** organize code into classes. A class combines variables and procedures into a single construct. Most languages developed since 1990 are object-oriented. Ex: Java, Python, and C++.

Declarative languages do not contain control flow statements. Each statement declares *what* result is desired, using logical expressions, rather than *how* the result is processed. Compilers for declarative languages are called **optimizers**, since the compiler determines an optimal way to process each declarative statement. SQL is the leading example of a declarative language.

Figure 8.1.1: Programming languages.



[Feedback?](#)**PARTICIPATION ACTIVITY**

8.1.1: Programming languages.



- 1) _____ languages combine data and procedures into classes.

Check[Show answer](#)**Correct** or

Object-oriented languages contain control flow statements but, unlike procedural languages, combined procedures and related variables into classes.



- 2) _____ languages do not explicitly specify how results are processed.

Check[Show answer](#)**Correct**

A declarative language contains no control flow statements and therefore does not specify processing steps.



- 3) _____ languages contain control flow statements but no classes.

Check[Show answer](#)**Correct**

Early programming languages, such as C, FORTRAN, and COBOL, are procedural and contain control flow statements. Object-oriented languages are composed of classes and became popular in the 1990s.



- 4) Programming languages are either _____ or declarative.

Check[Show answer](#)**Correct**

Programming languages that contain control flow statements are imperative. Programming languages that do not contain control flow statements are declarative.

[Feedback?](#)

Database programming

Declarative language is commonly used for database queries for two reasons:

- **Easier programming.** With procedural language, complex queries are coded with loops and conditional statements. With declarative language, complex queries are coded in just one statement.
- **Faster execution.** With procedural language, a programmer specifies how to process a query. With declarative language, the optimizer determines how to process a query. Since complex queries can be processed in many ways, optimized queries usually execute faster than programmed queries.

For the reasons above, as well as early backing by IBM, SQL has become the dominant query language.

Complete applications need control flow statements as well as database queries. Since SQL has no control flow statements, complete applications combine SQL with a procedural or object-oriented language. Building applications with both SQL and a general-purpose language is called ***database programming***.

Database programming presents two challenges:

- **Syntax gap.** The syntax of SQL and most other languages are quite different. Ex: Groups of statements, called blocks, are fundamental to most procedural and object-oriented languages. SQL has no blocks.
- **Paradigm gap.** Query processing is fundamentally different in SQL and procedural or object-oriented languages. Ex: A result set with many rows is processed in a single SQL statement, but usually requires loops and conditional statements in other languages.

Database programming must bridge the syntax and paradigm gaps.

PARTICIPATION
ACTIVITY

8.1.2: Syntax and paradigm gaps.



1 2 3 4 5 6 2x speed

Declarative language

```
SELECT AirlineName, FlightNumber
FROM Flight
WHERE AirportCode = 'JFK' AND DepartureTime > '12:00';
```

Procedural language

```
printf("%20s %12s \n", "AirlineName ", "FlightNumber");
for (int i = 1; i <= rowCount(flight); i++) {
    if (strcmp(getAirportCode(i), "JFK") == 0 && getDepartureTime(i) > 12)
        printf("%20s %12d \n", getAirlineName(i), getFlightNumber(i));
}
```

AirlineName	FlightNumber
American Airlines	1154

printf() displays the airline name and flight number of selected flights.

Captions 

1. A declarative language specifies what result is desired. The SQL selects all airline names and flight numbers departing from JFK airport after noon.
2. A procedural language specifies how the result is processed. The code fragment is written in C.
3. printf() displays column headings. The string containing % characters specifies the heading format.
4. The for loop checks every row of the Flight table. Variable i is the row number.
5. The if statement selects flights departing from JFK airport after noon.
6. printf() displays the airline name and flight number of selected flights.

[Feedback?](#)

PARTICIPATION
ACTIVITY

8.1.3: Database programming.



- 1) _____ is the leading declarative language.

SQL

[Check](#)

[Show answer](#)

Correct

SQL



Although many database processing languages are declarative, SQL was backed by IBM when relational database first became popular. As a result, SQL became the most widely used declarative language.

- 2) A(n) _____ determines how to process statements in a declarative language.

optimizer

[Check](#)

[Show answer](#)

Correct

optimizer or compiler



A declarative language specifies what results are desired. The compiler determines how the result is processed. Compilers that determine the fastest execution plan are called optimizers.

- 3) C contains control flow statements and SQL does not. This is an example of the _____ gap.

paradigm



Correct

paradigm

C is a procedural language. SQL is a declarative language. Procedural and declarative are different

Check**Show answer**

4) SQL uses many keywords, like SELECT, FROM, and WHERE. C relies heavily on punctuation, like {} and #. This is an example of the _____ gap.

syntax

Check**Show answer****Correct**

syntax

Use of keywords versus punctuation is a syntax difference, and does not necessarily create a different programming language paradigm.

**Feedback?**

Database programming techniques

To bridge the syntax and paradigm gaps, three database programming techniques have emerged:

- **Embedded SQL** codes SQL statements directly in a program written in another language. The keywords `EXEC SQL` precede all SQL statements so the compiler can distinguish SQL from other statements. Ex: Embedded SQL in C for Oracle Database is called Pro*C.
- **Procedural SQL** extends the SQL language with control flow statements, creating a new programming language. Procedural SQL is limited compared to general-purpose languages, however, and is used primarily for database applications. Ex: In Oracle Database, procedural SQL is called PL/SQL.
- An **application programming interface**, or **API**, is a library of procedures or classes. The library links an application programming language to a computer service, such as a database, email, or web service. The procedure or class declarations are written in the application programming language. Ex: JDBC is a library of Java classes that access relational databases.

Embedded SQL was the earliest database programming technique, developed along with SQL in the 1980s. Procedural SQL emerged soon after embedded SQL in the late 1980s. The first widely used database API, ODBC, was released in 1992.

Procedural SQL and APIs are often used together. A common database task is coded in an SQL procedure. The compiled procedure is called via an API and can be reused in many applications.

The animation below illustrates the C# object-oriented language using the ODBC database API.

**PARTICIPATION
ACTIVITY**

8.1.4: API example using C# and ODBC.



1 2 3 4 5 6   2x speed

```
string reservationConnString =  
    "DRIVER={MySQL ODBC 3.51 Driver};" +  
    "SERVER=localhost;" +  
    "DATABASE=Reservation;" +  
    "UID=samsnead;" +  
    "PASSWORD=e%8nH!";  
OdbcConnection reservationConnection = new OdbcConnection(reservationConnString);  
reservationConnection.Open();  
  
OdbcCommand listFlights = new OdbcCommand("SELECT * FROM Flight", reservationConnection);  
  
OdbcDataReader flightReader;  
flightReader = listFlights.ExecuteReader();  
  
while (flightReader.Read()) {  
    Console.WriteLine("Flight Number: " + flightReader.GetInt32(0));  
    Console.WriteLine("Airline Name: " + flightReader.GetString(1));  
}  
  
flightReader.Close();
```

Flight Number: 208
Airline Name: United Airlines
Flight Number: 920
Airline Name: Lufthansa

GetInt32(0) and GetString(1) return the first and second columns of the current FlightReader row.

Captions 

1. The reservationConnection object connects the program to the reservation database.
2. The listFlights object contains the SELECT statement and connects to the Reservation database.
3. The flightReader object points to individual rows of the result table.
4. ExecuteReader() executes the SELECT statement stored in the listFlights object.
5. Read() moves flightReader to the next result table row. The while loop repeats until all rows have been read.
6. GetInt32(0) and GetString(1) return the first and second columns of the current FlightReader row.

[Feedback?](#)

PARTICIPATION ACTIVITY

8.1.5: Database programming.



- 1) Different database programming techniques

Correct



- cannot be combined in one application.
- True
 False

- 2) The EXEC SQL keyword is used in procedural SQL.
- True
 False

- 3) Procedural SQL is specified in the SQL standard.
- True
 False

- 4) The ODBC API for the Java language consists of Java classes.
- True
 False

Applications often combine APIs with procedural SQL. With this approach, common data processing tasks can be coded once and reused in several applications.

Correct

EXEC SQL keyword is used in embedded SQL and distinguishes SQL statements from the host language statements.

Correct

Early versions of the SQL standard did not include control flow statements. The standard was extended, however, shortly after the first procedural SQL languages became available.

Correct

Procedure and class declarations for an API are always written in the host language. However, the procedures and classes may be implemented in a different language.

[Feedback?](#)

Advantages and disadvantages

Embedded SQL allows the programmer to write SQL statements directly within another programming language. This approach is conceptually simple but suffers three problems:

- **Gaps.** Most programming today uses object-oriented languages. The syntax and paradigm gap between object-oriented language and SQL is wide, so programs are difficult to write and maintain.
- **Compile steps.** The SQL statements and host language must be compiled with different compilers, in two steps.
- **Network traffic.** Application programs usually run on a client computer, connected to the database server via a network. With embedded SQL, each query generates a 'round-trip' from client to server, increasing network traffic and execution time.

For the above reasons, embedded SQL is no longer widely used.

Procedural SQL has several advantages over embedded SQL:

- **Gaps.** Procedural SQL gracefully extends SQL. The syntax and paradigm gaps are minimal so programs are relatively easy to write and maintain.
- **Compile steps.** Procedural SQL is compiled in one step rather than two.
- **Network traffic.** Client applications typically call SQL procedures for execution on the database server. Each procedure executes multiple database operations, reducing network traffic and execution time.
- **Optimization level.** The database optimizes entire procedures rather than individual queries, resulting in better optimization and faster query execution.

Procedural SQL languages vary significantly by database and do not offer the full capabilities of general-purpose languages. Ex: Most procedural SQL languages are not object-oriented. For these reasons, procedural SQL is commonly used for limited database tasks.

APIs address the limitations of embedded SQL and procedural SQL:

- **Gaps.** Since API procedures and class declarations are written in the application language, applications are compiled in one step, the syntax gap disappears, and the paradigm gap is reduced.
- **Applications.** Database APIs are available for general-purpose, object-oriented languages. Applications are not limited to database processing tasks and procedural languages.
- **Database independence.** Unlike procedural SQL, language and API syntax is independent of data source.

For the reasons above, APIs are the most commonly used database programming technique.

Table 8.1.1: Advantages and disadvantages.

	Paradigm gap	Syntax gap	Object-oriented language	Applications	Compile steps	Network traffic	Database-independent
Embedded SQL	High	High	Limited	General	2	High	No
Procedural SQL	Moderate	Low	Limited	Database processing	1	Low	No
API	Moderate	Low	Yes	General	1	Variable	Yes

[Feedback?](#)

PARTICIPATION ACTIVITY

8.1.6: Advantages and disadvantages.



Select the database programming technique that best matches the description.

- 1) Usually generates a network round trip for each SQL query.

- Embedded SQL
- Procedural SQL
- API

- 2) Compiles and stores complex data processing tasks on the database server.

- Embedded SQL
- Procedural SQL
- API

- 3) Has become less popular with the rise of object-oriented programming.

- Embedded SQL
- Procedural SQL
- API

- 4) Affords the greatest database security.

- Embedded SQL
- Procedural SQL
- API

Correct

Application programs usually run on client computers. When SQL is embedded in an application program, each query generates a round trip to the database server.

**Correct**

Procedural SQL usually codes several queries in one procedure. The procedure is compiled and stored on the database server, for use with many applications.

**Correct**

The SQL declarative paradigm is quite different than the object-oriented paradigm. Consequently, most database programming with object-oriented languages use an API.

**Correct**

With procedural SQL, complex tasks can be executed with one network round-trip. Individual steps of each task are executed on the server, which is less vulnerable than the network.

**Feedback?**

Exploring further:

- Declarative programming
- Procedural and object-oriented programming

How

was this
section?



[Provide feedback](#)