

DATABASE DESIGN & PROGRAMMING WITH SQL
COURSE CODE: 5324

COURSE DESCRIPTION: This curriculum is geared to meet the learning needs of a variety of students, from those interested in gaining broad exposure to business and technical skills to students planning on pursuing a technical education or career. This course utilizes an Oracle hosted, state-of-the-art lab environment to build database design and programming skills. Students analyze case studies to identify data patterns and connections in order to design relational databases. Students create entity relationship diagrams (ERDs) while building collaboration and problem solving skills. Students build and modify databases using structured query language (SQL), the industry-standard database programming language.

OBJECTIVE: Given the necessary equipment, supplies, and facilities, the student will complete all of the following core standards successfully.

RECOMMENDED GRADE LEVELS: 10-12

COURSE CREDIT: 1 or 2 unit(s)

PREREQUISITE: Basic computer skills (word processing, Internet use) and, recommended, successful completion of Algebra 1.

COMPUTER REQUIREMENT: one computer per student; Internet access; see Oracle Academy specifications for additional information

RECOMMENDED SOFTWARE: Adobe Acrobat Professional; Mozilla Firefox; Google Chrome

RESOURCES: Oracle iLearning online curriculum; Oracle Application Express; both provided by Oracle Academy

INDUSTRY CREDENTIALS/CERTIFICATIONS AVAILABLE:

The Oracle Database SQL Certified Expert certification is available by passing the Oracle Database SQL Expert Exam 1Z0-047 taken at a certified testing facility.

Students may begin the process for the Oracle PL/SQL Developer Certified Associate certification by passing either 1Z0-047 (described above) or Oracle Database 11g: SQL Fundamentals Exam 1Z0-051 (taken online) or Introduction to Oracle9i: SQL Exam 1Z0-007 (taken online).

Since Exam 1Z0-047 results in a recognized certification and must be taken at a certified testing facility, it is more expensive than Exam 1Z0-051 and Exam 1Z0-007. Discount coupons for exams may be available from Oracle Academy. Third-party, supplemental practice exams and materials are recommended, and may be available at discounted prices.

Certification requirements and pathways are subject to change and should be confirmed prior to exam registration.

A. SAFETY AND ETHICS

1. Identify major causes of work-related accidents in offices.
2. Describe the threat of viruses to a computer network, methods of avoiding attacks, and options in dealing with virus attacks.
3. Identify potential abuse and unethical uses of computers and networks.
4. Explain the consequences of illegal, social, and unethical uses of information technologies, e.g., piracy; illegal downloading; licensing infringement; and inappropriate uses of software, hardware, and mobile devices.
5. Differentiate between freeware, shareware, and public domain software copyrights.
6. Discuss computer crimes, terms of use, and legal issues such as copyright laws, fair use laws, and ethics pertaining to scanned and downloaded clip art images, photographs, documents, video, recorded sounds and music, trademarks, and other elements for use in Web publications.
7. Identify netiquette including the use of email, social networking, blogs, texting, and chatting.
8. Describe ethical practices in business professions such as safeguarding the confidentiality of business-related information.

B. EMPLOYABILITY SKILLS

1. Identify positive work practices, e.g., appropriate dress code for the workplace, personal grooming, punctuality, time management, organization.
2. Demonstrate positive interpersonal skills, e.g., communication, respect, and teamwork.

C. STUDENT ORGANIZATIONS

1. Explain how related student organizations are integral parts of career and technology education courses.
2. Explain the goals and objectives of related student organizations.
3. List opportunities available to students through participation in related student organization conferences/competitions, community service, philanthropy, and other activities.
4. Explain how participation in career and technology education student organizations can promote lifelong responsibility for community service and professional development.

DATABASE DESIGN

D. INTRODUCTION TO DATABASE DESIGN & PROGRAMMING

1. Distinguish the difference between data and information.
2. Explain how data becomes information.
3. Describe the evolution of the database.
4. Identify examples of databases in the business world.
5. Describe the database development process.
6. Identify examples of essential database use in e-businesses.

E. DATA MODELING

1. Distinguish between a database's conceptual model and its physical implementation.
2. Explain the significance of the four goals of entity relationship modeling.
3. Define an entity relationship diagram.
4. Define an entity.
5. Identify examples of an entity.
6. Distinguish between an entity and an instance of an entity.
7. Name and describe attributes for a given entity.
8. Distinguish between an attribute and its value.
9. Distinguish between mandatory and optional attributes, and between volatile and nonvolatile attributes.
10. Identify appropriate unique identifiers (UID) for various entities.

F. ENTITY RELATIONSHIP DIAGRAMMING

1. Construct ERD components that represent entities and attributes according to accepted diagramming conventions.
2. Explain relationship optionality.
3. Explain relationship cardinality.
4. State relationships between entities in precise words (ERDish).
5. Draw properly labeled relationships on an ERD.
6. Draw an ERD from a matrix diagram.

G. SUPERTYPES, SUBTYPES, AND BUSINESS RULES

1. Explain the relationship between supertypes and subtypes.
2. Create an ERD with a supertype and subtypes.
3. Compose structural and procedural business rules.
4. Diagram appropriate business rules in an ERD.

H. WORKING WITH ENTITY RELATIONSHIPS

1. Describe relationship transferability and nontransferability.
2. Illustrate nontransferable relationships on ERDs.
3. Define entity relationships (one-to-one, one-to-many, and many-to-many).
4. Identify redundant relationships.
5. Demonstrate the steps to resolve a many-to-many relationship using an intersection entity.
6. Identify the UID of an intersection entity and represent it on an ERD.
7. Identify the create, retrieve, update, and delete (CRUD) requirements of a simple business.
8. Validate an ERD by performing a CRUD analysis.

I. UNIQUE IDENTIFIERS AND NORMALIZATION

1. Define the different types of unique identifiers (UIDs) (artificial, composite, and secondary).
2. Explain the issues of real world identification of entity instances.
3. Define the purpose of normalization in database models.
4. Define the rules of First Normal Form, Second Normal Form, and Third Normal Form.
5. Apply the rules of First Normal Form, Second Normal Form, and Third Normal Form.

J. ARCS, HIERARCHIES, AND HISTORICAL DATA

1. Define the term "constraint" as it applies to data modeling.
2. Identify an exclusive OR relationship.
3. Diagram an arc constraint to represent an exclusive OR relationship.
4. Distinguish between the use of an arc and a subtype in a data model.
5. Define a hierarchical relationship.
6. Identify the UIDs in a hierarchical model.
7. Define a recursive relationship.
8. Construct an ERD using equivalent recursive and hierarchical relationships.
9. Identify the need to track data that changes over time.
10. Construct a model that incorporates elements of data over time.

K. MODELING CHANGE

1. Distinguish between using date as an attribute and DAY as an entity in a data model.
2. Identify at least three time-related constraints that can result from a time-sensitive model.
3. Identify examples of conditional non-transferability in a time-constrained model.

4. Define journaling or logging as it applies to tracking changes in value over time.
5. Construct a model that includes journaling or logging to track changes in value over time.
6. Correctly use DATE as an entity in an ERD.

L. DRAWING CONVENTIONS AND GENERIC MODELING

1. Identify high-volume entities in an ERD.
2. Use accepted diagramming conventions to modify an ERD to increase clarity and readability.
3. Define generic modeling.
4. Describe the advantages and disadvantages of generic modeling.
5. Construct an ERD using generic modeling conventions.

M. TRANSFORMING CONCEPTUAL MODEL INTO PHYSICAL MODEL

1. Define a primary key.
2. Define a foreign key.
3. Define a column-integrity rule.
4. Identify row, column, primary key, unique key, and foreign key from a table diagram.
5. Identify violations of data-integrity rules.
6. Transform simple and complex ERDs into database tables, columns, keys, and constraints using specific naming conventions.

N. INTRODUCTION TO SQL PROGRAMMING

1. Use CREATE command to create a database table.
2. Use INSERT command to insert data into a table.
3. Create a basic SELECT statement to validate successful data insertion.
4. Use correct syntax to display all columns and rows.
5. Use correct syntax to display specified columns and rows.
6. Modify a table's structure by adding a new column.
7. Use DELETE and ALTER TABLE commands to correct mistakes or make revisions to a table.
8. Describe the different stages of the system-development life cycle.
9. Identify the role of data modeling in the system-development life cycle.

O. PROJECT DEVELOPMENT

1. Demonstrate effective intra-team communication.
2. Track team project progress.
3. Compose well-organized written documentation.
4. Prepare visuals that support the database documentation.
5. Demonstrate the use of a central message and supporting arguments for a business presentation.

6. Demonstrate logical analysis of the business rules, operations, and processes in a way that is clear and easy for the client to understand.
7. Demonstrate selection of appropriate attire for a business presentation.
8. Demonstrate appropriate use of eye contact, upright posture, gestures, and other nonverbal communication during a business presentation.
9. Demonstrate clear and concise answers to all questions during a business presentation.

P. SELECT STATEMENTS AND DATABASE TECHNOLOGY

1. Distinguish between application software and system software.
2. Identify features of the Oracle 11g Database Environment.
3. Demonstrate use of Oracle Application Express Version 3.2 to run SQL statements that retrieve data.
4. Use the correct syntax to modify the way data is displayed.
5. Use the correct syntax to display the results of calculations using arithmetic expressions and operators.
6. Match projection, selection, and join with their correct functions/capabilities.

Q. WHERE CLAUSES

1. Apply the concatenation operator to link columns to other columns, arithmetic expressions, or constant values to create a character expression.
2. Use column aliases to rename columns in the query result.
3. Use literal values of type character, number, or date in a SELECT statement.
4. Use DISTINCT to eliminate duplicate rows.
5. Display the structure of a table using DESCRIBE or DESC.
6. Save SQL statements in Oracle Application Express.
7. Demonstrate appropriate use of the WHERE clause.
8. Create a SQL query containing character strings and date values.
9. Apply the proper comparison operator to return a desired result.
10. Use BETWEEN, IN, and LIKE to return a desired result.
11. Explain the value of NULL.
12. Explain the use of comparison conditions and NULL.

R. RESTRICTING ROWS AND INTRODUCTION TO FUNCTIONS

1. Evaluate logical comparisons to restrict the rows returned based on two or more conditions.
2. Apply the rules of precedence to determine the order in which expressions are evaluated and calculated.
3. Construct a query to display results in ascending and descending order.

4. Construct a query to order the results using a column alias.
5. Construct a query to order the results for single and multiple columns.
6. Identify appropriate applications of single-row functions.
7. Classify a function as a single-row or multi-row function.
8. Differentiate between single-row functions and multi-row functions.

S. INTRODUCTION TO INFORMATION TECHNOLOGY CAREERS

1. Explore IT careers by evaluating personal goals, IT careers, expected skills, and required training.
2. Analyze IT career options based on interests, skills, and achievements and/or aptitude and interests surveys.
3. Prepare a resume.

PROGRAMMING WITH SQL

T. CHARACTER, NUMBER, AND DATE FUNCTIONS

1. Use character case-manipulation functions LOWER, UPPER, and INITCAP in SQL queries.
2. Use character-manipulation functions CONCAT, SUBSTR, LENGTH, INSTR, LPAD, RPAD, TRIM, and REPLACE in SQL queries.
3. Write flexible queries using substitution variables.
4. Use single-row number functions ROUND, TRUNC, and MOD in SQL queries.
5. Explain the business implications of the different results when applying TRUNC and ROUND to numeric values.
6. Use single-row functions MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, and TRUNC in SQL queries.
7. Explain how date functions transform dates into date data or a numeric value.
8. Demonstrate proper use of the arithmetic operators with dates.
9. Demonstrate the use of SYSDATE.

U. SINGLE ROW FUNCTIONS

1. Explain explicit data-type conversion and implicit data-type conversion.
2. Explain why it is important, from a business perspective, for a programming language to have built-in data-conversion capabilities.
3. Construct a SQL query that correctly applies TO_CHAR, TO_NUMBER, and TO_DATE single-row functions to produce a desired result.
4. Apply the appropriate date and/or character format model to produce a desired output.
5. Use YYYY and RRRR to return the correct year.
6. Explain the result of nested functions.
7. List at least four general functions that work with any data type and relate to handling null values.

8. Explain the use of the COALESCE and the NVL functions.
9. Explain the use of general functions to deal with null values in data.
10. Create SQL queries that use NVL, NVL2, NULLIF, and COALESCE single-row functions.
11. Explain the difference between the DECODE and CASE functions.
12. Create SQL queries that use the DECODE and CASE functions.
13. Construct two methods for implementing IF-THEN-ELSE conditional logic.

V. TABLE JOINS

1. Create a natural join using ANSI-99 SQL join syntax
2. Create a cross join using ANSI-99 SQL join syntax
3. Define the relationship between a cross join and a Cartesian product
4. Define the relationship between a natural join and an equijoin.
5. Explain why it is important to have the ANSI standard for SQL.
6. Describe a business need for combining data from multiple tables to create useable information.
7. Create a join using ANSI-99 USING and ON clauses.
8. Create an ANSI-99 query that joins three tables.
9. Explain the difference between an inner and an outer join.
10. Use a left outer join, a right outer join, and a full outer join in SQL queries.
11. Create a SELECT statement to join a table to itself.
12. Create a tree-structured report.
13. Format hierarchical data.

W. GROUP FUNCTIONS

1. Use group functions SUM, AVG, COUNT, MIN, MAX, STDDEV, and VARIANCE in SQL queries.
2. Use DISTINCT and NVL functions with group functions.

X. COMPLEX SQL WITH AGGREGATED DATA

1. Create SQL statements using nested group functions.
2. Use GROUP BY and GROUP BY ... HAVING in SQL queries.
3. Use ROLLUP to produce subtotal values.
4. Use CUBE to produce cross-tabulation values.
5. Use GROUPING SETS to produce a single result set.
6. Use the GROUPING function to identify the extra row values created by either a ROLLUP or CUBE operation.
7. Use a SET operator to combine multiple queries into a single query.
8. Use SET operator to control the order of rows returned.

Y. SUBQUERIES

1. Explain the purpose of subqueries for retrieving data.

2. Distinguish between single-row and multiple-row subqueries.
3. Distinguish between pair-wise and non-pair-wise subqueries.
4. Use a single-row subquery in a WHERE clause.
5. Use EXIST and NOT EXISTS operators in a query.
6. Identify when correlated subqueries are needed.
7. Use correlated subqueries in SQL queries.
8. Create named subqueries using the WITH clause.

Z. DML STATEMENTS

1. Explain why it is important to be able to alter data in a database.
2. Create INSERT statements which insert a single row using a VALUES clause.
3. Create INSERT statements that use special values, null values, and date values.
4. Create INSERT statements that copy rows from one table to another using a subquery.
5. Use UPDATE and DELETE to modify a database.
6. Use a subquery and a correlated subquery to update and delete data.
7. Explain how foreign-key and primary-key integrity constraints affect UPDATE and DELETE statements.
8. Explain when to specify a DEFAULT value.
9. Use a MERGE statement to modify data.
10. Execute DML statements using subqueries.
11. Execute multi-table inserts.

AA. DDL STATEMENTS

1. Use SQL commands to query the Data Dictionary to identify database objects, their attributes, and their structures.
2. Explain the use of external tables.
3. Create a table using TIMESTAMP and TIMESTAMP WITH TIME ZONE column data types.
4. Create a table using INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND column data types.
5. Give examples of organizations and personal situations where it is important to know to which time zone a date-time value refers.
6. Explain why it is important to be able to modify a table structure.
7. Explain the effect on tables and columns of the DDL statements ALTER, DROP, RENAME, and TRUNCATE.
8. Use the ALTER TABLE commands ADD, MODIFY, and DROP.
9. Explain FLASHBACK table operations.
10. Perform FLASHBACK QUERY on a table.
11. Track the changes to data over a period of time.
12. Explain the rationale for using TRUNCATE versus DELETE for tables.
13. Add a comment to a table using the COMMENT ON TABLE command.
14. Explain when the SET UNUSED statement is advantageous.

BB. VALIDATING QUERY RESULTS

1. Use validation methodology to modify queries to produce the expected results.

CC. CONSTRAINTS

1. Define the term "constraint" as it relates to data integrity.
2. Explain when a constraint must or may be added at the column level and when it must or may be added at the table level.
3. State why it is important to give meaningful names to constraints.
4. Identify data integrity enforced by NOT NULL and UNIQUE constraints.
5. Write CREATE TABLE statements that include NOT NULL and UNIQUE constraints at the table and column levels.
6. Explain how constraints are created at the time of table creation.
7. Explain what is constrained by a PRIMARY KEY, a FOREIGN KEY, and a CHECK constraint.
8. Explain the purpose of defining PRIMARY KEY, FOREIGN KEY, and CHECK constraints.
9. Write CREATE TABLE statements that include constraints at the column level and table level.
10. Evaluate a business problem requiring the addition of a PRIMARY KEY and FOREIGN KEY constraint.
11. Interpret the information returned when you query the Data Dictionary for USER_CONSTRAINTS.
12. List the functions the ALTER statement can perform on constraints.
13. Write ALTER TABLE statements to add, drop, disable, and enable constraints.
14. Name a business function that would require a DBA to drop, enable, and/or disable a constraint, or use the CASCADE syntax.

DD. VIEWS

1. List three uses for views from the standpoint of a DBA.
2. Explain why it is important for a business to be able to create and use logical subsets of data derived from one or more tables.
3. Create a view with and without column aliases in the subquery using a single base table.
4. Create a complex view that contains group functions to display values from two tables.
5. Retrieve data from a view.
6. Write a query that performs DML operations on a simple view.
7. Identify the conditions that restrict your ability to modify a view using DML operations.
8. Write a query that includes a WITH CHECK OPTION clause.
9. Explain the use of WITH CHECK OPTION as it applies to integrity constraints and data validation.
10. Use the WITH READ ONLY option in a view to restrict DML operations.

11. Create a SQL statement that removes a view.
12. Create a query to create an inline view.
13. Create a top-n-analysis query.

EE. SEQUENCES

1. List the benefits of using a sequence.
2. Write a SQL statement that creates a sequence.
3. Query the Data Dictionary using USER_SEQUENCES to confirm a sequence definition.
4. Use NEXTVAL to generate sequential unique numbers in a table.
5. List the advantages and disadvantages of caching sequence values.
6. Identify how gaps can occur in a sequence.
7. Explain the use of an index as a schema object.
8. Explain the use of ROWID in locating information in a database.
9. Identify the conditions that automatically create an index.
10. Use CREATE INDEX and DROP INDEX statements.
11. Create a function-based index.
12. Create a private and a public synonym.

FF. DATABASE SECURITY

1. Explain the difference between object privileges and system privileges.
2. Write SQL syntax to grant database access to a new user.
3. Create a GRANT...ON...TO statement to assign privileges to objects in your schema to other users.
4. Query the Data Dictionary to confirm the granting of privileges.
5. Explain the advantage of creating a ROLE.
6. Construct a statement to create a ROLE and GRANT privileges to it.
7. Construct a statement to REVOKE a user's object.
8. Explain the difference between privileges and roles.
9. Explain the purpose of a database link.
10. Describe regular expressions.
11. Use regular expressions to search, match, and replace strings in SQL statements.
12. Use regular expressions with check constraints.

GG. DATABASE TRANSACTIONS

1. Define the terms COMMIT, ROLLBACK, and SAVEPOINT as they relate to data transactions.
2. List the advantages of using COMMIT, ROLLBACK, and SAVEPOINT statements.
3. Explain why it is important to control the flow of transaction processing.

HH. ORACLE PROPRIETARY JOINS

1. Identify the Oracle proprietary joins and their ANSI/ISO SQL: 1999 counterparts.
2. Create a SELECT statement that results in a Cartesian product.
3. Create a SELECT statement to access data from more than one table using Oracle equijoin syntax.
4. Create SELECT statements that add search conditions using AND.
5. Use column aliases in a join statement.
6. Create a SELECT statement to access data from more than one table using Oracle nonequijoin syntax.
7. Create a SELECT statement to access data from more than one table using Oracle outer join syntax.

II. DATABASE DEVELOPMENT

1. Use the Application Builder tool in Oracle Application Express to create a database for a small business by creating tables, layouts, pages, list of values, forms, and reports.
2. Implement a strategy to test that a database functions as designed.

JJ. ADVANCED QUERY RESULTS

1. Create an advanced query to produce specified data.
2. Modify an advanced query to produce the specified data.

KK. CAREER PLANNING

1. Identify trends in the current job market.
2. Identify employers of interest and compatibility.
3. Write a cover letter and an interview follow-up letter.
4. Write a letter requesting a letter of recommendation.
5. Discuss formal and informal career networking.
6. Identify job search techniques.
7. Explain the different types of interviews.
8. Demonstrate appropriate behavior and dress for an interview.
9. Ask appropriate questions in a mock interview.
10. Identify risky behavior within an online social network that could be detrimental to your career.