

The background features a dark gray map with white lines representing streets and building footprints. On the right side, there is a stylized graphic of a building facade composed of a grid of white squares of varying sizes, creating a 3D effect.

ConstructionMasters

A Database Systems Case Study

PROJECT OVERVIEW

LOGICAL DESIGN

Table of volumes,
Access tables,
Redundancy Analysis,
Logical Schema

WEB APPLICATION

Servlets and HTML
pages developed
with Eclipse and
Tomcat



CONCEPTUAL DESIGN

Structuring of phrases,
Glossary of terms,
Conceptual schema,
Business rules

PHYSICAL DESIGN

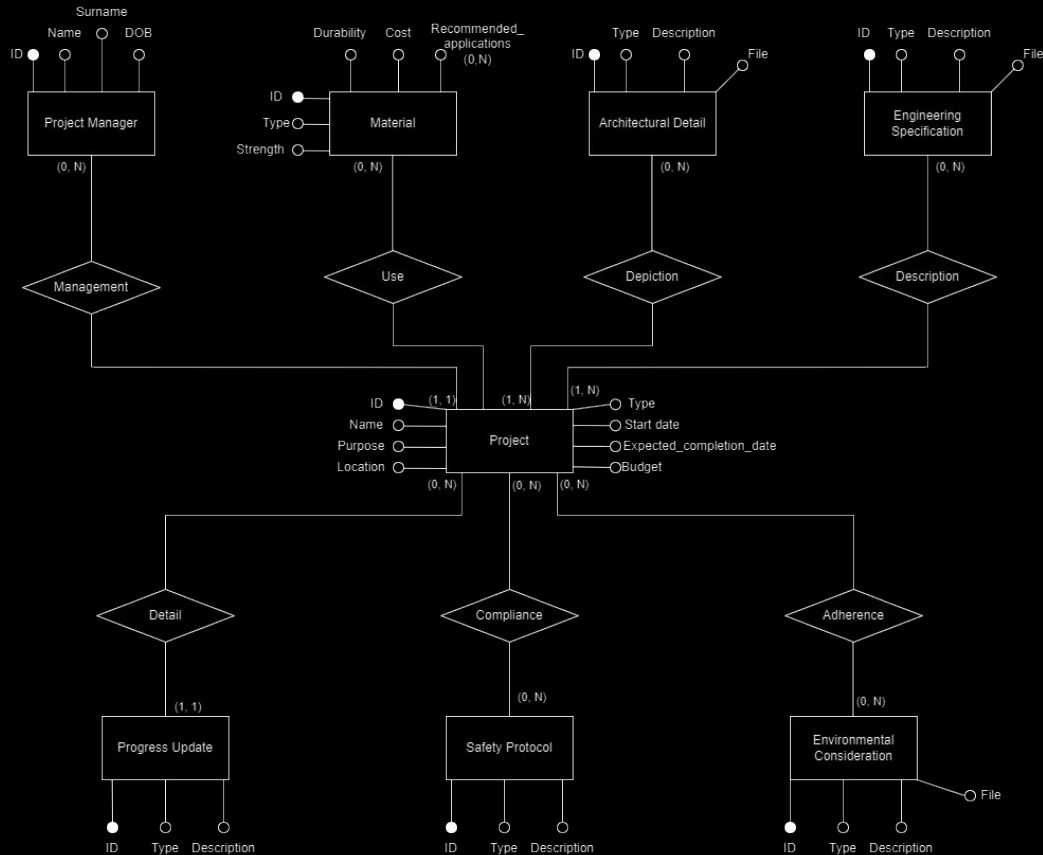
Types and Tables,
Triggers, Procedures
and functions, Query
optimization



GLOSSARY OF TERMS

TERM	DESCRIPTION	SYNONYMS	LINKS
Project	Construction endeavor of a specific type (residential, commercial, industrial).	Construction Project	Project Manager, Material, Architectural Detail, Engineering Specification, Progress Update, Safety Protocol
Project Manager	Responsible for overseeing the project.	Manager	Project
Material	Element used in construction (concrete, steel, wood, glass).	Construction elements, Construction materials	Project
Architectural Detail	Architectural designs including building designs, blueprints, floor plans, 3D models.	-	Project
Engineering Specification	Structural designs, load-bearing capacities, safety measures.	-	Project
Progress Update	Milestones achieved and challenges faced in a project.	-	Project
Safety Protocol	Safety measures and regulations for the project.	-	Project
Environmental Consideration	Promotes eco-friendly construction methods.	-	Project

CONCEPTUAL SCHEMA



BUSINESS RULES

01

The 'Type' attribute of the 'Project' entity must be 'residential', 'commercial' or 'industrial'.

02

The 'Type' attribute of the 'Architectural Detail' entity must be 'building design', 'blueprint', 'floor plan', or '3D model'.

03

The 'Type' attribute of the 'Engineering Specification' entity must be 'structural design', 'load-bearing capacity', or 'safety measure'.

04

The 'Type' attribute of the 'Safety Protocol' entity must be 'standard' or 'requirement'.

05

The 'Type' attribute of the 'Environmental Consideration' entity must be 'construction practice', 'design', or 'green material'.

06

A new 'Project Manager' can be inserted if their DOB indicates they are 18 years old or older.

07

A project's expected completion date must always be after the start date.

08

The total cost of materials for a project cannot exceed the budget of the project.

LOGICAL DESIGN



TABLE OF VOLUMES

Some volumes were based on specifications, while others were deduced by making some assumptions.



REDUNDANCY ANALYSIS

A new (redundant) attribute, 'number of milestones', was introduced to reduce the cost of operation 6 from 210'008 to 10'008.



ACCESS TABLES

Estimation of the cost of the described operations on the database.

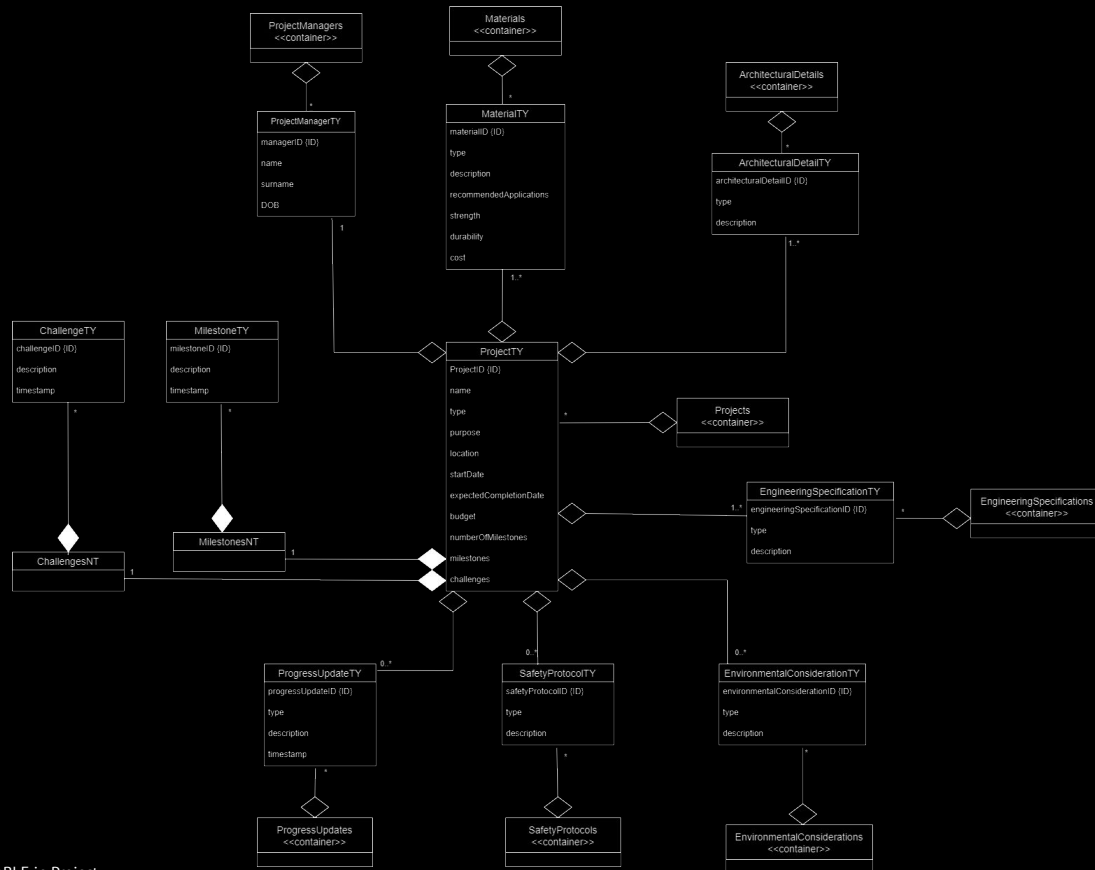
OP6 (without redundancy)

Concept	Type	Accesses	Type	Explanation
Project	E	10000	R	Read all projects
Detail	R	100000	R	Assumption: each project has an average of 10 progress updates
Progress Update	E	100000	R	Assumption: each project has an average of 10 progress updates, need to access all of them to find the milestones
Depiction	R	2	R	Assumption: Each project has 2 architectural details on average
Architectural Detail	E	2	R	Assumption: Each project has 2 architectural details on average
Description	R	2	R	Assumption: Each project has 2 engineering specifications on average
Engineering Specification	E	2	R	Assumption: Each project has 2 engineering specifications on average

OP6 (with redundancy)

Concept	Type	Accesses	Type	Explanation
Project	E	10000	R	Read all projects
Depiction	R	2	R	Assumption: Each project has 2 architectural details on average
Architectural Detail	E	2	R	Assumption: Each project has 2 architectural details on average
Description	R	2	R	Assumption: Each project has 2 engineering specifications on average
Engineering Specification	E	2	R	Assumption: Each project has 2 engineering specifications on average

LOGICAL SCHEMA



*error: ProgressUpdateTY should have have a NESTED TABLE in Project (composition)

TYPES AND TABLES

TYPES

ProjectManagerType
MaterialType
ArchitecturalDetailType
EngineeringSpecificationType
ProgressUpdateType
SafetyProtocolType
EnvironmentalConsiderationType
MilestoneType
ChallengeType
ProjectType

TABLES

ProjectManager
Material
ArchitecturalDetail
EngineeringSpecification
SafetyProtocol
EnvironmentalConsideration
ProgressUpdate
Project

JUNCTION TABLES

ProjectMaterial
ProjectArchitecturalDetail
ProjectEngineeringSpecification
ProjectSafetyProtocol
ProjectEnvironmentalConsideration
ProjectProgressUpdate

TRIGGERS



ENFORCING TYPES

project_type_check
architectural_detail_type_check
safety_protocol_type_check
environmental_consideration_type_check



MANAGER AGE

ProjectManagerDOBCheck



PROJECT DURATION

project_date_check
(the expected end date must follow
the start date)



BUDGET VS. COST

check_budget
(the cost of the materials used in the
project must not exceed the budget)



FUNCTIONS



insert_new_project

Creates a new project with the specified attributes



print_project_and_milestones

Retrieves and prints information about a given project and its milestones.



print_total_material_cost

Computes and prints the total cost of construction materials for a specific project



print_projects_and_challenges

Retrieves and prints information about all projects and their associated challenges, sorted by project start date.



print_projects_by_material_cost

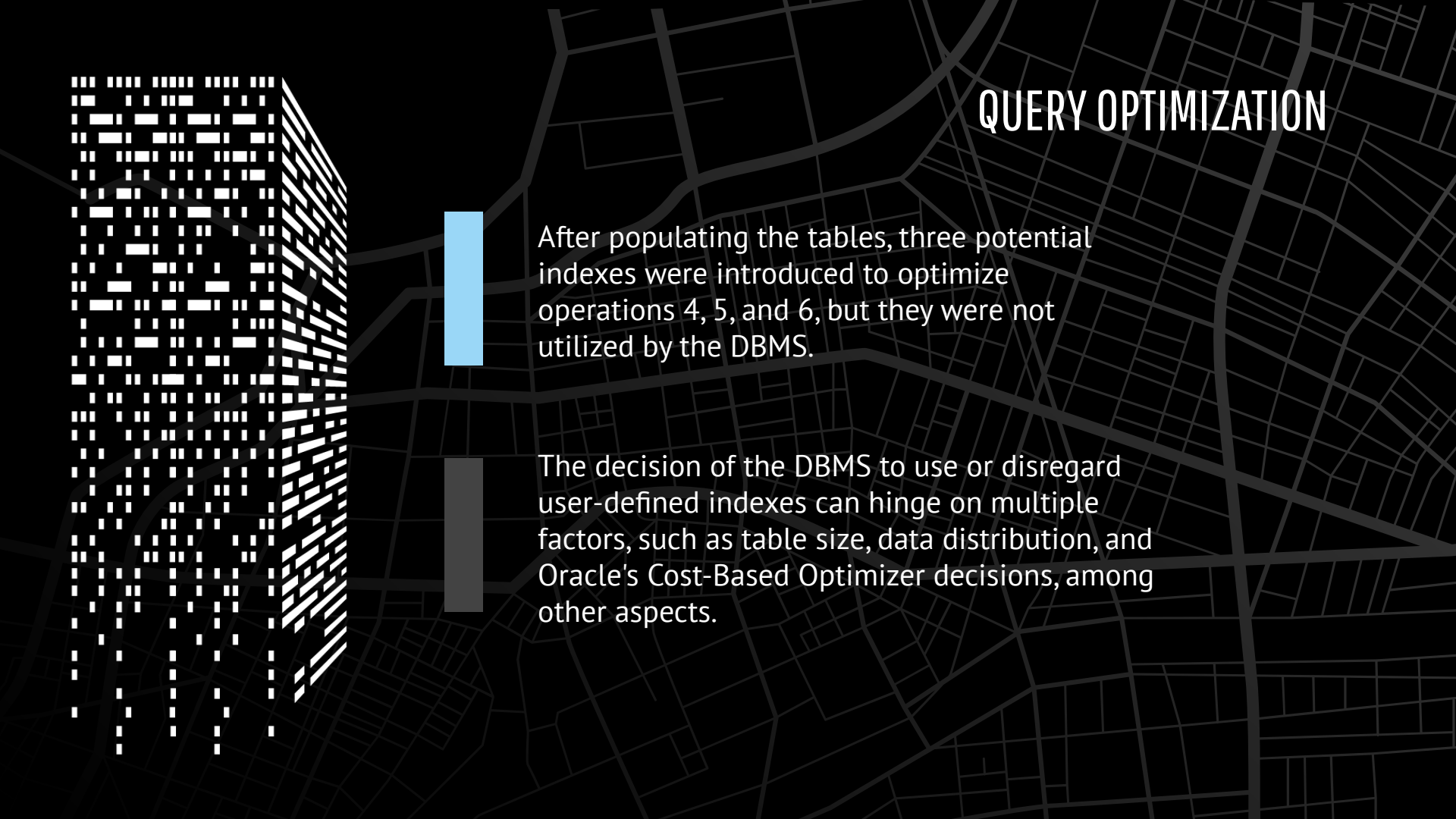
Retrieves and prints information about all projects, sorted by the total cost of construction materials used in descending order.



print_lowest_milestone_project_details

Retrieves and prints information about the architectural and engineering features and progress updates for the project with the fewest milestones.

QUERY OPTIMIZATION



After populating the tables, three potential indexes were introduced to optimize operations 4, 5, and 6, but they were not utilized by the DBMS.

The decision of the DBMS to use or disregard user-defined indexes can hinge on multiple factors, such as table size, data distribution, and Oracle's Cost-Based Optimizer decisions, among other aspects.

ProjectServlet

Implements operation number 1: inserting a new project into the database. Calls the stored procedure `insert_new_project`.

Browser

Insert New Project

Project ID:

Name:

Type:

Purpose:

Location:

Start Date:

Expected Completion Date:

Budget:

Number of Milestones:

Manager ID:

Tomcat Server

ProjectServlet

```
public class ProjectServlet extends HttpServlet {  
    private static final long serialVersionUID = 80489863762089860L;  
  
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        String url = request.getHeader("Location");  
        String url = "system";  
        String password = "oracle";  
  
        try {  
            Class.forName("oracle.jdbc.driver.OracleDriver");  
            Connection con = DriverManager.getConnection(url, user, password);  
  
            CallableStatement stmt = con.prepareCall("{call insert_new_project(?, ?, ?, ?, ?, ?, ?, ?, ?)}");  
            stmt.setString(1, request.getParameter("projectID"));  
            stmt.setString(2, request.getParameter("name"));  
            stmt.setString(3, request.getParameter("type"));  
            stmt.setString(4, request.getParameter("purpose"));  
            stmt.setString(5, request.getParameter("location"));  
            stmt.setString(6, request.getParameter("startDate"));  
            stmt.setString(7, request.getParameter("expectedCompletionDate"));  
            stmt.setString(8, request.getParameter("budget"));  
            stmt.setString(9, request.getParameter("numberOfMilestones"));  
            stmt.setString(10, request.getParameter("managerID"));  
            stmt.execute();  
            response.getWriter().println("Project inserted successfully");  
            stmt.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Browser

Project inserted successfully!

PROJECTID	NAME	TYPE	PURPOSE
987	987 yexqSYWktIwLzrDhYgoA	residential	qkSXhVoZhgjsobaoONRwFXPvcgawAY
988	988 qInhtIdPLeDsraJUohxa	residential	bGfyYhRdoryLaR0Q:ClFOhKnMghA
989	989 GUurQPpTsnS0BhSvpnC	residential	fALWENqNnpwLwniPQiGAgurufec
990	990 tUcrqviDwlrjwpUJG	residential	sluAXGZjQIhndneqccQWNBGLJPVlr
991	991 AvbTMYBGGrHxGUmQXyz	residential	zWtCs0wNB1JsdttWc:ZgZiZiungPe
992	992 uFywykttAbqrNvVBSenF	residential	duqVobmarLbVLk1BdAdsrXfUmVI
993	993 qUUMWpJEaFaZYFXERQUL	residential	ukvthWFRkynSeexqQGBDeziPhoTz
994	994 NJPDbjIFMrbMaESNZMH	residential	uKJyCoRyPPvGZTV0sSViabpShRzZ
995	995 OPFvxiGiDzeTFQDdBtOJ	residential	ZsfiTfQbXIZWmChNkHkHqGRhJqN
996	996 tdHhLW1LTOJwvdJmPLLE	residential	NNBxRRNjawaQscFJhawyVlAKVTxiQ
997	997 anQINFGshQvquMwNj	residential	YWTXyEtFabuzVewLbTZeLddPuvJm
998	998 OssVamZhdigazexaTsr	residential	FXOLJrrTaOwgfgQFSzRzLUwKIr
999	999 kvaMzLWFLZKvRFRcaJIR	residential	eJLJOGzBvauhp0Fsdga11CjheYvhJ
1000	1000 UCnAntFzVvUgeNusC	residential	HdEchMDKEBFSzeewRpwzJfvvgYuw
1001	2345 New Project	commercial	luxury apartment complex

ProjectInfoServlet



Implements operation 2: retrieves and prints information about a given project and its milestones.



Unlike the first servlet, it does not directly call a stored procedure.



Instead, it reproduces the logic of `print_project_and_milestones` within the servlet itself due to the limitations of the `DBMS_OUTPUT` package.

Browser

Project Information

Project ID:

Browser

```
Project ID: 5, Project Name: RICHPEBLKJLUREBIN, Milestone Description:
16qpho50Y0Y0qW6Lc1sAMWATZ4FlPHbVoaLpIacRgRg
Project ID: 5, Project Name: RICHPEBLKJLUREBIN, Milestone Description:
JkVtZclpPmPggqGamb1qCWMPdXLLJUBctChwKpGmbCh
Project ID: 5, Project Name: RICHPEBLKJLUREBIN, Milestone Description:
fp0duWzkmvyaW0ZthYrQwheZjAZcBwanRydgzPMxydRPO
Project ID: 5, Project Name: RICHPEBLKJLUREBIN, Milestone Description:
AdeTstONwiczjYlrmJGsxzWqU5craSqFXYVpPqPHQXBEM
```



MaterialCostServlet

Implements operation number 3: calculating and displaying the total material cost for a specified project ID.

Browser

Material Cost Information

Project ID:

Calculate Cost

Browser

Project ID: 100, Total Material Cost: 241

Tomcat Server



MaterialCostServlet

```
public class MaterialCostServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
        String url = "jdbc:oracle:thin@localhost:1521/orcl";
        String user = "system";
        String password = "oracle";

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection conn = DriverManager.getConnection(url, user, password);

            String sql = "SELECT SUM(m.cost) as totalcost FROM Project p JOIN ProjectMaterial pm ON p.prj_id = pm.prj_id";
            PreparedStatement stat = conn.prepareStatement(sql);
            stat.setInt(1, Integer.parseInt(request.getParameter("projectID")));
            ResultSet rs = stat.executeQuery();

            PrintWriter out = response.getWriter();
            if(rs.next()){
                out.println("Project ID: " + request.getParameter("projectID") + ", Total Material Cost: " + rs.getDouble(1));
            } else {
                out.println("No data found for project ID: " + request.getParameter("projectID"));
            }

            rs.close();
            stat.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println(e);
        }
    }
}
```

Oracle DB

ProjectChallengesServlet

Implements operation 4: displays the challenges associated with each project.

Browser

Project Challenges

Load Challenges

Browser

Project ID: 762, Project Name: cTxCYfNysZifbCsQ0TKD, Challenge Description: RGoMBzKSLXPZicETTPiEfXmVjTJNnKJlPzKudzNwxxjwAJTMGF
Project ID: 667, Project Name: oKINlPAEvndLQwVweWTS, Challenge Description: qUHLficEubpIdtfySrTFRXEWjMDyqxXuoQVkvbgaoowHuaJP
Project ID: 795, Project Name: fXkjsQyJfeSxvTyczh, Challenge Description: QwGURAdoPKiFMULtZdzgeHPLuDrfrqEgIdoxXwvTLoyBwOtn
Project ID: 235, Project Name: f3KxY0tHzwvIdaGccIR, Challenge Description: ooiCJBqVGamZNSDkrqCQJemzNNGMNgEGDcqoloEIgZheJLI
Project ID: 970, Project Name: EhhdicNygifaFPeUXbp, Challenge Description: VALBBnLtldGpZBARfLhSiaNqbvRLrktHUTLABNtpFGYGDwtmH
Project ID: 644, Project Name: ttRstbiMorAutgvbOvOV, Challenge Description: pLVK0enuTsEJeJFWfWeBbh0oowJINNoJHXzuKXaEhuxLjXSe
Project ID: 80, Project Name: LpLixjPKg6fVUvNema, Challenge Description: TypelepryubHjzTuPaqHfYfSdAYwUdgetjXpVcexIH0XoPy
Project ID: 217, Project Name: HkHgwOVTVwAUKlMrIbt, Challenge Description: jIxcYKhGOWjpzJkBXzkrqDCJMgXmvKPEtENBwsbizgCKOHNRetJ
Project ID: 794, Project Name: ssbarQlfjKSghNSuAaMH, Challenge Description: DwMxLbrJvTAfIAOXGXGnCYMZouYhXjaXnTSOAYKPFcXIKsFRH
Project ID: 794, Project Name: ssbarQlfjKSghNSuAaMH, Challenge Description: HGSpfnfcrzhK5kyUcQvTLSEUZTpQvGIPFvPwPHVYtANpTO
Project ID: 43, Project Name: HA7aHGFvSPDOTzjVnbaP, Challenge Description: TUBGFFAPkQxRPDUqSzAvpWMANIDPNAOTcCEXLfGwHBNKCYLT
Project ID: 966, Project Name: cwhVlKnYSDamsifKxsw, Challenge Description: iPrbSNEsyIncmTDxPdFvbingePKZsiuWpZPMaVrOP10fbPp
Project ID: 938, Project Name: LBS0YKyZLiTGzmbuFwLq, Challenge Description: MCFsdrkaxkaEeEUokqQmLwMLITgErigdQMqQrgVAVTXFDYnu

ProjectChallengesServlet

```
public class ProjectChallengesServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,  
        String url = "jdbc:oracle:thin@localhost:1521/orcl";  
        String user = "system";  
        String password = "oracle";  
  
        try {  
            Class.forName("oracle.jdbc.driver.OracleDriver");  
            Connection conn = DriverManager.getConnection(url, user, password);  
  
            String sql = "SELECT p.projectID, p.name, c.description FROM Project p, TABLE(p.challenges) c";  
            PreparedStatement stat = conn.prepareStatement(sql);  
            ResultSet rs = stat.executeQuery();  
  
            PrintWriter out = response.getWriter();  
            while(rs.next()) {  
                out.println("Project ID: " + rs.getInt("projectID") + ", Project Name: " + rs.getString("name") + ", Challenge Description: " + rs.getString("description") + "<br>");  
            }  
  
            rs.close();  
            stat.close();  
            conn.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

Tomcat Server



Oracle DB

HighestCostServlet

Implements operation number 5: fetches the projects sorted by their total material cost.

Browser

Projects by Material Cost

Show Projects

Browser

Project ID: 50, Project Name: 0E1a3dtdx0frUcLb0ok, Total Material Cost: 2042.3158502900739
Project ID: 323, Project Name: ep0WqaoynfqXuxFpL, Total Material Cost: 2012.541776764218
Project ID: 649, Project Name: E0WY2FyLYFis4DCW0iQ, Total Material Cost: 1967.4939763151635
Project ID: 366, Project Name: NLoj5ftf3h0bKp0wU, Total Material Cost: 1916.1509414135613
Project ID: 891, Project Name: r0fL3QwXy1cZfCp0v, Total Material Cost: 1915.1102617579145
Project ID: 98, Project Name: I1hpaXc0F0n6Crxp0, Total Material Cost: 1896.17919403444
Project ID: 546, Project Name: mWcRfH0a0QJx7lB0p, Total Material Cost: 1875.4694054060911
Project ID: 329, Project Name: Gw0S0P0t0r0P0m0y0, Total Material Cost: 1871.92871296992
Project ID: 698, Project Name: I1hVpUfXPKYl0ntJhV1K, Total Material Cost: 1857.181448257656
Project ID: 840, Project Name: w0gh1R0Jvdf0u0w0jH, Total Material Cost: 1846.331210809919
Project ID: 799, Project Name: f0k0d0h0j0k0G1c0n0L, Total Material Cost: 1840.88102137806
Project ID: 519, Project Name: tE0pHSE1sR0R00R0h, Total Material Cost: 1840.532303831944
Project ID: 741, Project Name: G0apJH50vZU0M0q0PH, Total Material Cost: 1828.4155211690939
Project ID: 499, Project Name: A0By0K1c0n0X0u0p, Total Material Cost: 1817.000515786377
Project ID: 155, Project Name: NqfX0KZv0l7h0APvALR, Total Material Cost: 1804.542156838534
Project ID: 825, Project Name: S0wL0G0 seZ0D0MUV0P, Total Material Cost: 1776.7411617625941
Project ID: 535, Project Name: J0I2TEG00pL1s0u0P, Total Material Cost: 1766.0417002784108
Project ID: 890, Project Name: f0xstv1I0ks0ntk0rZ, Total Material Cost: 1762.5383097108781
Project ID: 597, Project Name: t0M0Jy0D0n0f0r1l2p0, Total Material Cost: 1755.9356605968223
Project ID: 886, Project Name: kv0c1j0g0S0u0R0t0w, Total Material Cost: 1737.1108632158276
Project ID: 771, Project Name: 0t0nky0u0b0nky0s0u0P, Total Material Cost: 1717.9362288024527
Project ID: 415, Project Name: h0V0uV0q1y0w1zENB0K, Total Material Cost: 1710.9386651224631
Project ID: 113, Project Name: s7Y0K0X0L0N0Y0h0d0A0N, Total Material Cost: 1702.816820214904
Project ID: 224, Project Name: T0u0ed0PZu0R0L0Hy0K, Total Material Cost: 1699.9026683506152
Project ID: 988, Project Name: q1nH1d1p1Ed0S0J0h0x, Total Material Cost: 1693.1495324782452
Project ID: 47, Project Name: G0B7C0d0L0c0C0w0r1f, Total Material Cost: 1674.1389797430262

Tomcat Server



HighestCostServlet

```
public class HighestCostServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        String url = "jdbc:oracle:thin:@localhost:1521/orcl";
        String user = "system";
        String password = "oracle";

    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        connection = DriverManager.getConnection(url, user, password);

        String sql = "SELECT p.projectID, p.name, SUM(m.cost) as totalMaterialCost " +
            "FROM Project p " +
            "JOIN ProjectMaterial pm ON p.projectID = pm.projectID " +
            "JOIN Material m ON pm.materialID = m.materialID " +
            "GROUP BY p.projectID, p.name";

        PreparedStatement stmt = connection.prepareStatement(sql);
        ResultSet rs = stmt.executeQuery();

        PrintWriter out = response.getWriter();
        while(rs.next()){
            out.println("Project ID: " + rs.getInt("projectID") + ", Project Name: " + rs.getString("name"));
        }
        rs.close();
        stmt.close();
        connection.close();
    } catch (Exception e) {
        e.printStackTrace();
        out.println(e);
    }
}
```

Oracle DB

LowestMilestoneProjectServlet

Implements operation number 6: displaying the architectural details, engineering specifications, and progress updates of the project with the fewest milestones.

