UdemyDB Database Documentation

**Introduction**

The **UdemyDB** database is designed to manage an online learning platform, supporting functionalities like user management, course creation, enrollment tracking, progress monitoring, and analytics. This documentation covers **85 stored procedures** and **40 views** that power core operations, divided into **4 categories** of stored procedures (*Get, Modify, Input, Remove*) and **3 types of views** (*Dimension, Fact, Utility*). The system emphasizes security, scalability, and analytics readiness, with Power BI integration for reporting.

**Stored Procedures (85)**

Organized by functional categories and their **key features**:

**1. Get Procedures (Select Operations)**

**Purpose**: Retrieve data with filtering, sorting, and analytics-ready formatting.  
**Key Features**:

* **Filtering**: Supports parameters like @MinRating, @CourseId, or @IncludeDeleted for granular queries.
* **JSON Support**: Returns structured data (e.g., CourseGoals as JSON arrays).
* **Progress Tracking**: Calculates metrics like ProgressPercentage and CompletionRate.
* **Hierarchical Data**: Formats sections/lessons hierarchically for frontend rendering.
* **Role-Based Data**: Separates student/instructor views with role checks.

**Critical Procedures**:

|  |  |
| --- | --- |
| **Procedure** | **Key Functionality** |
| sp\_GetInstructorINFOByID | Returns instructor earnings, social links, and aggregated course stats. |
| sp\_GetCourseDetailsById | Combines course metadata, goals, requirements, and instructor social links in one call. |
| sp\_GetStudentProgress | Tracks lesson completion, course progress %, and generates progress heatmaps. |
| sp\_GetAllInstructorsINFO | Filters instructors by rating/course count and returns JSON-formatted course lists. |

**2. Modify Procedures (Update Operations)**

**Purpose**: Update records with transaction safety and cascading changes.  
**Key Features**:

* **Transaction Handling**: Uses BEGIN TRANSACTION and ROLLBACK for data integrity.
* **Auto-Recalculations**: Updates derived fields (e.g., CurrentPrice after discount changes).
* **Status Propagation**: Changes like course progress trigger enrollment status updates.
* **Validation**: Checks user roles, enrollment status, or soft-delete flags before updates.

**Critical Procedures**:

|  |  |
| --- | --- |
| **Procedure** | **Key Functionality** |
| sp\_UpdateCourse | Recalculates discounts and IsFree flag based on price changes. |
| sp\_UpdateLessonProgress | Updates lesson status and cascades to course-level progress percentages. |
| sp\_CompleteOrderAndEnroll | Converts cart to enrollment, handling partial failures with error logging. |
| sp\_UpdateCourseRating | Updates student ratings and refreshes course average ratings. |

**3. Input Procedures (Insert Operations)**

**Purpose**: Insert records with validation and initialization.  
**Key Features**:

* **Atomic Operations**: Creates dependent records (e.g., carts for new students).
* **Role Initialization**: Assigns roles via AspNetUserRoles and initializes student/instructor profiles.
* **Duplicate Checks**: Prevents duplicate enrollments or cart entries.
* **Auto-Generated Fields**: Sets CreatedDate, IsDeleted=0, and initializes counts (e.g., TotalCourses=0).

**Critical Procedures**:

|  |  |
| --- | --- |
| **Procedure** | **Key Functionality** |
| sp\_CreateUser | Creates users with multiple roles and initializes profiles/carts in a single transaction. |
| sp\_AddLessonToSection | Adds lessons and auto-updates section duration/lesson counts. |
| sp\_CreateOrderFromCart | Converts cart items to orders while preserving historical prices. |
| sp\_AnswerQuestion | Links answers to questions with user validation. |

**4. Remove Procedures (Delete Operations)**

**Purpose**: Soft/hard delete records with cleanup logic.  
**Key Features**:

* **Soft Delete Pattern**: Uses IsDeleted flags for auditability (default).
* **Cascading Deletes**: Removes dependent records (e.g., deleting a course removes sections/lessons).
* **Resource Cleanup**: Frees unused resources (e.g., deleting carts releases student wallets).
* **Hard Delete Control**: Requires explicit @HardDelete=1 for irreversible operations.

**Critical Procedures**:

|  |  |
| --- | --- |
| **Procedure** | **Key Functionality** |
| sp\_DeleteUserAndAllData | Recursively deletes user-related data across 10+ tables with transaction safety. |
| sp\_SoftDeleteEnrollment | Removes enrollments and resets course subscriber counts. |
| sp\_DeleteCourse | Cascades deletions to sections, lessons, and quizzes. |
| sp\_RemoveCourseFromCart | Updates cart totals and auto-deletes empty carts. |

**Views (40)**

Designed for reporting, analytics, and Power BI integration:

**1. Dimension Views**

**Purpose**: Provide descriptive attributes for analytics.  
**Key Views**:

|  |  |
| --- | --- |
| **View** | **Description** |
| VDimCourse | Course attributes (Category, Price, Duration). |
| VDimUser | User demographics, roles, and social media presence. |
| VSupDimQuiz | Quiz structure (question types/counts) for difficulty analysis. |
| VSupDimSection | Section-level metadata (video/article counts) for content audits. |

**2. Fact Views**

**Purpose**: Track measurable events/metrics.  
**Key Views**:

|  |  |
| --- | --- |
| **View** | **Description** |
| vw\_FactEnrollment | Enrollment metrics (progress %, grades) with date keys for trend analysis. |
| vw\_FactOrder | Sales data (revenue, discounts) linked to students/courses. |
| vw\_FactCart | Cart activity (abandonment rates, course popularity). |

**3. Utility Views**

**Purpose**: Simplify complex queries for business reporting.  
**Key Views**:

|  |  |
| --- | --- |
| **View** | **Description** |
| vw\_CourseRevenue | Estimates revenue per course with instructor earnings (70% share). |
| vw\_StudentLearningActivity | Tracks student engagement (active courses, learning hours). |
| vw\_InstructorPerformance | Aggregates instructor KPIs (total students, courses, ratings). |
| vw\_CourseCompletionRates | Calculates completion % per course for quality audits. |

**Conclusion**

The UdemyDB stored procedures and views provide:

1. **Security**: Role checks, parameterized queries, and transaction safety.
2. **Scalability**: Efficient indexing on CourseId, StudentId, and InstructorId.
3. **Analytics Readiness**: Power BI-compatible views with star schema design.
4. **Maintainability**: Soft delete patterns and cascading operations.