**Assignment 02**  
**Group Members: Ibrahim, Mahmoud - Orozco Morales, Rodrigo - Yusuf, Sharmarke**   
 **Application description**

The movie database is designed to manage and store comprehensive information about movies, their classifications, and associated metadata. It is built using Oracle PL/SQL and consists of six interrelated tables: MOVIES, GENRES, COUNTRIES, LANGUAGES, CERTIFICATES, and DIRECTORS, along with a junction table MOVIES\_GENRES to handle the many-to-many relationship between movies and genres.

**Tables and Their Purpose:**

1. MOVIES: The core table that stores movie details. It includes:
   * ID (Primary Key): Unique identifier for each movie.
   * NAME, YEAR, RATING, RUN\_TIME, BUDGET, BOX\_OFFICE: Basic movie attributes.
   * GENRE\_ID, CERTIFICATION\_ID, DIRECTORS\_ID, COUNTRY\_ID, LANGUAGE\_ID: Foreign keys linking to other tables.
2. GENRES: Stores movie genre information.
   * GENRE\_ID (Primary Key): Unique identifier for each genre.
   * GENRE\_TYPE, DESCRIPTION: Genre details.
3. COUNTRIES: Stores country information.
   * COUNTRY\_ID (Primary Key): Unique identifier for each country.
   * COUNTRY\_NAME: Name of the country.
4. LANGUAGES: Stores language information.
   * LANGUAGE\_ID (Primary Key): Unique identifier for each language.
   * LANGUAGE\_NAME, NOTES: Language details.
5. CERTIFICATES: Stores movie certification/rating information (e.g., PG-13, R).
   * CERTIFICATE\_ID (Primary Key): Unique identifier for each certification.
   * NAME, DESCRIPTION: Certification details.
6. DIRECTORS: Stores director information.
   * DIRECTORS\_ID (Primary Key): Unique identifier for each director.
   * NAME, WORKS\_COUNT, AWARDS\_COUNT: Director details.
7. MOVIES\_GENRES: A junction table to manage the many-to-many relationship between MOVIES and GENRES.
   * MOVIE\_ID, GENRE\_ID: Composite primary key linking a movie to multiple genres.

**Relationships:**

* MOVIES to COUNTRIES: One-to-Many relationship. A movie is produced in one country (COUNTRY\_ID), but a country can be associated with many movies.
* MOVIES to LANGUAGES: One-to-Many relationship. A movie has one language (LANGUAGE\_ID), but a language can be used in many movies.
* MOVIES to CERTIFICATES: One-to-Many relationship. A movie has one certification (CERTIFICATION\_ID), but a certification can apply to many movies.
* MOVIES to DIRECTORS: One-to-Many relationship. A movie has one director (DIRECTORS\_ID), but a director can direct many movies.
* MOVIES to GENRES: Many-to-Many relationship. A movie has many genres (Associated Table), and a genre can have many movies.

**Additional Features:**

A view (MOVIES\_DETAILS\_VIEW) is created to join MOVIES with GENRES, COUNTRIES, LANGUAGES, and other tables to display comprehensive movie details (e.g., genre names instead of IDs). An INSTEAD OF trigger (MOVIES\_DETAILS\_TRG) is implemented on the view to handle INSERT, UPDATE, and DELETE operations, ensuring changes are reflected in the MOVIES table. The database is linked to Microsoft Access via ODBC, where a form allows users to manage movie records interactively.

**Purpose:**

The database enables efficient management of movie data, supports complex queries (e.g., finding movies by genre or director), and ensures data integrity through well-defined relationships and constraints. It is designed for scalability and ease of use in applications like movie catalog systems.