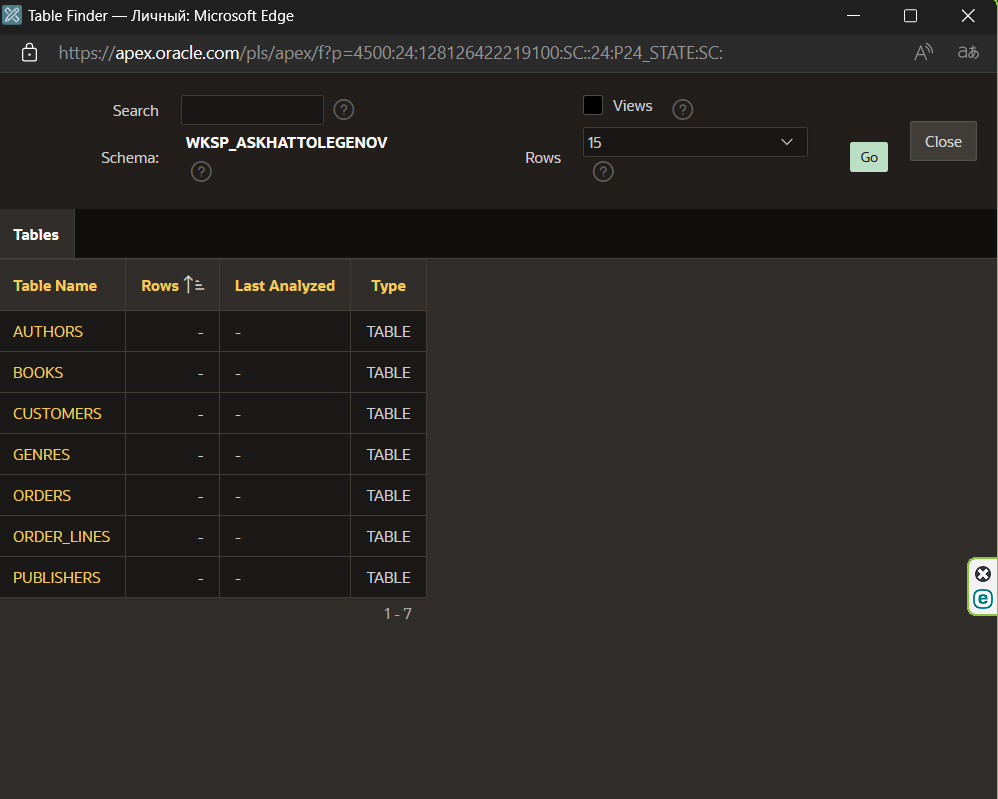


ER DIAGRAM



TABLES

PROCEDURE

CREATE OR REPLACE PROCEDURE GROUP\_BY\_BOOKS IS

AUTHOR\_ID VARCHAR2(100);

BOOK\_COUNT NUMBER;

AVG\_PRICE NUMBER;

BEGIN

SELECT AUTHOR\_ID, COUNT(\*) AS BOOK\_COUNT, AVG(PRICE) AS AVG\_PRICE

INTO AUTHOR\_ID, BOOK\_COUNT, AVG\_PRICE

FROM BOOKS

WHERE AUTHOR\_ID = 1

GROUP BY AUTHOR\_ID;

DBMS\_OUTPUT.PUT\_LINE('Author: ' || AUTHOR\_ID || ', Book Count: ' || BOOK\_COUNT || ', Avg Price: ' || AVG\_PRICE);

END;

FUNCTION

CREATE OR REPLACE FUNCTION COUNT\_ORDERS RETURN NUMBER

IS

OREDER\_COUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO OREDER\_COUNT FROM ORDERS;

RETURN OREDER\_COUNT;

END;

PROCEDURE

CREATE OR REPLACE PROCEDURE UPDATE\_BOOKS(

NEW\_ID IN NUMBER,

NEW\_TITLE IN VARCHAR2,

NEW\_AUTHOR\_ID IN VARCHAR2

)

IS

BEGIN

UPDATE BOOKS

SET TITLE = NEW\_TITLE,

AUTHOR\_ID = NEW\_AUTHOR\_ID

WHERE BOOK\_ID = NEW\_ID;

DBMS\_OUTPUT.PUT\_LINE('Number of rows updated: ' || SQL%ROWCOUNT);

END;

PROCEDURE

CREATE OR REPLACE PROCEDURE EXEPTION\_ADDBOOK(

NEW\_TITLE IN VARCHAR2,

NEW\_AUTHOR\_ID IN VARCHAR2,

NEW\_PUBLISHER\_ID IN VARCHAR2

)

IS

-- Define user-defined exception

TITLE\_ERROR EXCEPTION;

BEGIN

IF LENGTH(NEW\_TITLE) < 5 THEN

RAISE TITLE\_ERROR;

ELSE

INSERT INTO BOOKS(TITLE, AUTHOR\_ID, PUBLISHER\_ID)

VALUES (NEW\_TITLE, NEW\_AUTHOR\_ID, NEW\_PUBLISHER\_ID);

DBMS\_OUTPUT.PUT\_LINE('New book added successfully!');

END IF;

EXCEPTION

WHEN TITLE\_ERROR THEN

DBMS\_OUTPUT.PUT\_LINE('Book title must be at least 5 characters long!');

END;

TRIGGER

CREATE OR REPLACE TRIGGER TRIGGER\_ROWCOUNT

BEFORE INSERT ON BOOKS

FOR EACH ROW

DECLARE

ROW\_COUNT NUMBER;

BEGIN

SELECT COUNT(\*) INTO ROW\_COUNT FROM BOOKS;

DBMS\_OUTPUT.PUT\_LINE('Current row count: ' || ROW\_COUNT);

END;