

I. Porządkowanie wczytanych danych

1. Tworzenie tabel

```
CREATE TABLE TEAM
(
    TEAM_ID NUMBER(3) PRIMARY KEY,
    DESCRIPTION VARCHAR2(50)
);
```

Tabela TEAM będzie zawierała wygenerowane przez sekwencję identyfikatory zespołu (TEAM_ID) – będzie to klucz główny oraz nazwy zespołów (DESCRIPTION).

```
CREATE TABLE LOCATION
(
    LOCATION_ID NUMBER(3) PRIMARY KEY,
    LOCATION_NAME VARCHAR(100),
    CITY VARCHAR2(50),
    COUNTRY VARCHAR2(50)
);
```

Tabela LOCATION będzie zawierała wygenerowane identyfikatory lokalizacji (LOCATION_ID) – klucz główny, odpowiadające im nazwy lokalizacji (LOCATION_NAME) oraz nazwy lokalizacji podzielone na miasto i państwo (CITY, COUNTRY).

```
CREATE TABLE JOB
(
    JOB_ID NUMBER(3) PRIMARY KEY,
    LOCATION_ID NUMBER(3) REFERENCES LOCATION(LOCATION_ID),
    TEAM_ID NUMBER(3) REFERENCES TEAM(Team_ID),
    TITLE VARCHAR2(100),
    CREATED DATE
);
```

Tabela JOB będzie zawierała identyfikatory stanowisk pracy (JOB_ID) – klucz główny; identyfikatory lokalizacji, w której dane stanowiska się znajdują (LOCATION_ID) – klucz obcy, odnoszący się do tabeli LOCATION; identyfikatory zespołu, który jest przypisany do danego stanowiska (TEAM_ID) – klucz obcy, odnoszący się do tabeli TEAM; nazwy stanowisk pracy (TITLE) oraz daty, zmieniające się przy modyfikacji lub dodawaniu nowego rekordu do tabeli (CREATED).

2. Tworzenie sekwencji i wyzwalaczy

Tworzymy sekwencje, które będą generowały liczby przypisywane później przez wyzwalacze do tabel.

```
CREATE SEQUENCE SEQ_TEAM_ID;  
CREATE SEQUENCE SEQ_LOCATION_ID;  
CREATE SEQUENCE SEQ_JOB_ID;
```

Aby dodawać identyfikatory do tabel, tworzymy wyzwalacze:

```
CREATE OR REPLACE TRIGGER SET_TEAM_ID  
BEFORE INSERT  
ON TEAM  
FOR EACH ROW  
BEGIN  
    :NEW.TEAM_ID := SEQ_TEAM_ID.NEXTVAL;  
END;  
  
CREATE OR REPLACE TRIGGER SET_LOCATION_ID  
BEFORE INSERT  
ON LOCATION  
FOR EACH ROW  
BEGIN  
    :NEW.LOCATION_ID := SEQ_LOCATION_ID.NEXTVAL;  
END;  
  
CREATE OR REPLACE TRIGGER SET_JOB_ID  
BEFORE INSERT  
ON JOB  
FOR EACH ROW  
BEGIN  
    :NEW.JOB_ID := SEQ_JOB_ID.NEXTVAL;  
END;
```

Każdy z nich przypisuje wartość generowaną z sekwencji do pola z identyfikatorami w tabelach (TEAM, LOCATION, JOB) przed każdym dodawaniem rekordów do tych tabel.

```
CREATE OR REPLACE TRIGGER SET_DATE  
BEFORE INSERT OR UPDATE OF TITLE  
ON JOB  
FOR EACH ROW  
BEGIN  
    :NEW.CREATED := SYSDATE();  
END;
```

Wyzwalacz ten dodaje aktualną datę za każdym razem, gdy rekordy są dodawane do tabeli JOB lub modyfikuje się pole z nazwą stanowiska (TITLE).

3. Tworzenie pakietu pozwalającego na modyfikację bazy

```
137 CREATE OR REPLACE PACKAGE PACK AS
138   PROCEDURE INSERT_FROM_TEMP;
139 FUNCTION COUNT_JOB_OFFERS (
140   U_CITY LOCATION.CITY%TYPE,
141   U_TEAM_NAME TEAM.DESCRPTION%TYPE
142 ) RETURN NUMBER;
143 FUNCTION MAX_AMOUNT_OF_JOB_OFFERS
144   RETURN VARCHAR2;
145 END;
146 /
147 CREATE OR REPLACE PACKAGE BODY PACK AS
148 PROCEDURE INSERT_FROM_TEMP IS
149 BEGIN
150   MERGE INTO TEAM tea
151   USING ( SELECT DISTINCT TEAM
152         FROM TEMP1
153         ORDER BY 1 ) t
154   ON (tea.DESCRPTION = t.TEAM)
155   WHEN NOT MATCHED THEN
156     INSERT(DESCRPTION) VALUES (t.TEAM);
157
158   MERGE INTO LOCATION loc
159   USING ( SELECT DISTINCT LOCATION,
160         SUBSTR(LOCATION, 1, INSTR(LOCATION, ',')-1) AS CITY,
161         SUBSTR(LOCATION, INSTR(LOCATION, ',')+2) AS COUNTRY
162         FROM TEMP1
163         ORDER BY 1 ) l
164   ON (loc.CITY = l.CITY AND loc.COUNTRY = l.COUNTRY)
165   WHEN NOT MATCHED THEN
166     INSERT(LOCATION_NAME, CITY, COUNTRY) VALUES (l.LOCATION, l.CITY, l.COUNTRY);
167
168   MERGE INTO JOB jo
169   USING ( SELECT l.LOCATION_ID,
170         t.TEAM_ID,
171         temp.JOB_TITLE
172         FROM TEMP1 temp JOIN LOCATION l ON temp.LOCATION = l.LOCATION_NAME
173         JOIN TEAM t ON temp.TEAM = t.DESCRPTION
174         ORDER BY 1, 2, 3 ) j
175   ON (jo.LOCATION_ID = j.LOCATION_ID AND jo.TEAM_ID = j.TEAM_ID)
176   WHEN NOT MATCHED THEN
177     INSERT(LOCATION_ID, TEAM_ID, TITLE)
178     VALUES (j.LOCATION_ID, j.TEAM_ID, j.JOB_TITLE);
179 END INSERT_FROM_TEMP;
180
181 FUNCTION COUNT_JOB_OFFERS
182 ( U_CITY LOCATION.CITY%TYPE,
183   U_TEAM_NAME TEAM.DESCRPTION%TYPE
184 ) RETURN NUMBER IS AMOUNT NUMBER;
185
186 BEGIN
187   SELECT COUNT(j.JOB_ID)
188   INTO AMOUNT
189   FROM JOB j JOIN LOCATION l ON j.LOCATION_ID = l.LOCATION_ID
190         JOIN TEAM t ON j.TEAM_ID = t.TEAM_ID
191   WHERE U_CITY = l.CITY AND
192         U_TEAM_NAME = t.DESCRPTION;
193   RETURN AMOUNT;
194 END COUNT_JOB_OFFERS;
195
196 FUNCTION MAX_AMOUNT_OF_JOB_OFFERS
197 RETURN VARCHAR2 IS CITY VARCHAR2(32767);
198
199 BEGIN
200   SELECT l.CITY
201   INTO CITY
202   FROM LOCATION l, JOB j
203   WHERE j.LOCATION_ID = l.LOCATION_ID
204   GROUP BY l.CITY
205   HAVING COUNT(j.LOCATION_ID) = ( SELECT MAX(COUNT(j.LOCATION_ID))
206         FROM LOCATION l, JOB j
207         WHERE j.LOCATION_ID = l.LOCATION_ID
208         GROUP BY l.CITY );
209   RETURN CITY;
210 END MAX_AMOUNT_OF_JOB_OFFERS;
211
212 END PACK;
```

Od 137. linijki do 145. deklarujemy specyfikację pakietu. Nadajemy mu nazwę (PACK) i wypisujemy, co się w nim znajduje (jedna procedura i dwie funkcje). Następnie tworzymy ciało pakietu (linijki 147-221) W tym miejscu wpisujemy instrukcje inicjalizujące pakietu poprzedzone ich definicjami.

Pierwszym podprogramem pakietu jest procedura INSERT_FROM_TEMP (148-179). Dodaje ona dane do tabel TEAM, LOCATION, JOB dane z tabeli TEMP. W tabeli LOCATION dodatkowo zostaje rozdzielona kolumna z tabeli TEMP na dwie kolumny: CITY i COUNTRY.

Drugim podprogramem jest funkcja COUNT_JOB_OFFERS (181-194). Zlicza ona liczbę ofert pracy dla podanego miasta i zespołu (są to parametry wejściowe).

Ostatnim podprogramem jest funkcja MAX_AMOUNT_OF_JOB_OFFERS (196-210). Nie posiada ona parametrów wejściowych i zwraca miasto z największą liczbą ofert pracy.

II. Wywołanie procedury i funkcji z pakietu

1. Procedura INSERT_FROM_TEMP

```
EXECUTE PACK.INSERT_FROM_TEMP;
```

2. Funkcja COUNT_JOB_OFFERS z parametrami: Amsterdam oraz Finance

```
SELECT PACK.COUNT_JOB_OFFERS('Amsterdam', 'Finance') RESULT  
FROM DUAL;
```

Otrzymujemy wynik:

	RESULT
1	9

3. Funkcja MAX_AMOUNT_OF_JOB_OFFERS

```
SELECT PACK.MAX_AMOUNT_OF_JOB_OFFERS RESULT  
FROM DUAL;
```

Otrzymujemy wynik:

	RESULT
1	Amsterdam

III. Aktywacja wyzwalacza SET_DATE

Wyzwalacz aktywował się za pierwszym razem przy wywołaniu funkcji INSERT_FROM_TEMP, przypisał wtedy daty 22/03/26 dla każdego wiersza kolumny CREATED.

JOB_ID	LOCATION_ID	TEAM_ID	TITLE	CREATED
1	1	1	1 Client Manager, FinTech	22/03/26
2	2	1	1 Client Manager, FinTech	22/03/26

Dla sprawdzenia wyzwalacza przy modyfikacji rekordu, redefiniuję wyzwalacz ustawiając nowe rekordy na datę następnego dnia:

```
:NEW.CREATED := SYSDATE()+1;
```

Piszę polecenie modyfikujące rekord z kolumny TITLE dla pierwszego wiersza. Znajduje się tam „Client Manager, FinTech”. Zamieniam to na „Client Manager” poleceniem:

```
UPDATE JOB  
SET TITLE = 'Client Manager'  
WHERE JOB_ID = 1;
```

W wyniku nasza tabela wygląda następująco:

JOB_ID	LOCATION_ID	TEAM_ID	TITLE	CREATED
1	1	1	Client Manager	22/03/27
2	2	1	Client Manager, FinTech	22/03/26

IV. Fragmenty posortowanych tabel

1. **SELECT * FROM LOCATION ORDER BY CITY;**

LOCATION_ID	LOCATION_NAME	CITY	COUNTRY
1	1 Amsterdam, Netherlands	Amsterdam	Netherlands
2	2 Amsterdam, Netherlands and 1 more	Amsterdam	Netherlands and 1 more
3	3 Amsterdam, Netherlands and 2 more	Amsterdam	Netherlands and 2 more
4	4 Amsterdam, Netherlands and 3 more	Amsterdam	Netherlands and 3 more
5	5 Amsterdam, Netherlands and 4 more	Amsterdam	Netherlands and 4 more
6	6 Athens, Greece	Athens	Greece
7	7 Atlanta, GA, United States and 5 more	Atlanta	GA, United States and 5 more
8	8 Auckland, New Zealand	Auckland	New Zealand
9	9 Austin, TX, United States	Austin	TX, United States
10	10 Austin, TX, United States and 7 more	Austin	TX, United States and 7 more

2. **SELECT * FROM TEAM ORDER BY DESCRIPTION;**

TEAM_ID	DESCRIPTION
1	12 Business Development & Support
2	10 Customer Service
3	2 Data Science & Analytics
4	4 Design & User Experience
5	8 Engineering
6	5 Finance
7	7 Finance & Legal
8	1 IT Services
9	13 Leadership
10	11 Legal & Public Affairs

3. **SELECT * FROM JOB ORDER BY TITLE;**

JOB_ID	LOCATION_ID	TEAM_ID	TITLE	CREATED
1	524	38	12 Month Contract Recruiter - Customer Service	22/03/26
2	395	6	12 Account Executive - Athens	22/03/26
3	399	9	12 Account Executive - Austin	22/03/26
4	411	13	12 Account Executive - Berlin (m/f/d)	22/03/26
5	415	16	12 Account Executive - Bristol	22/03/26
6	424	18	12 Account Executive - Budapest	22/03/26
7	428	22	12 Account Executive - Dublin	22/03/26
8	431	24	12 Account Executive - Edinburgh	22/03/26
9	435	27	12 Account Executive - Hamburg (m/f/d)	22/03/26
10	440	31	12 Account Executive - Johannesburg (Portuguese Speaker)	22/03/26