ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

Факултет по изчислителна техника и автоматизация

Катедра „КНТ“

СЕМЕСТРИАЛНА ДОМАШНА РАБОТА

по дисциплината „База данни”

на тема: „Управление на персонал във фирма ”

Изготвила: Виктория Викторова Костадинова

Проверил: преп. Антоанета Иванова Иванова-Димитрова

Специалност: Изкуствен интелект

Група: 1а

Факултетен номер: 23621913

1. Тема на проекта

Управление на служители, отсъствия и причини за отсъствия в организация

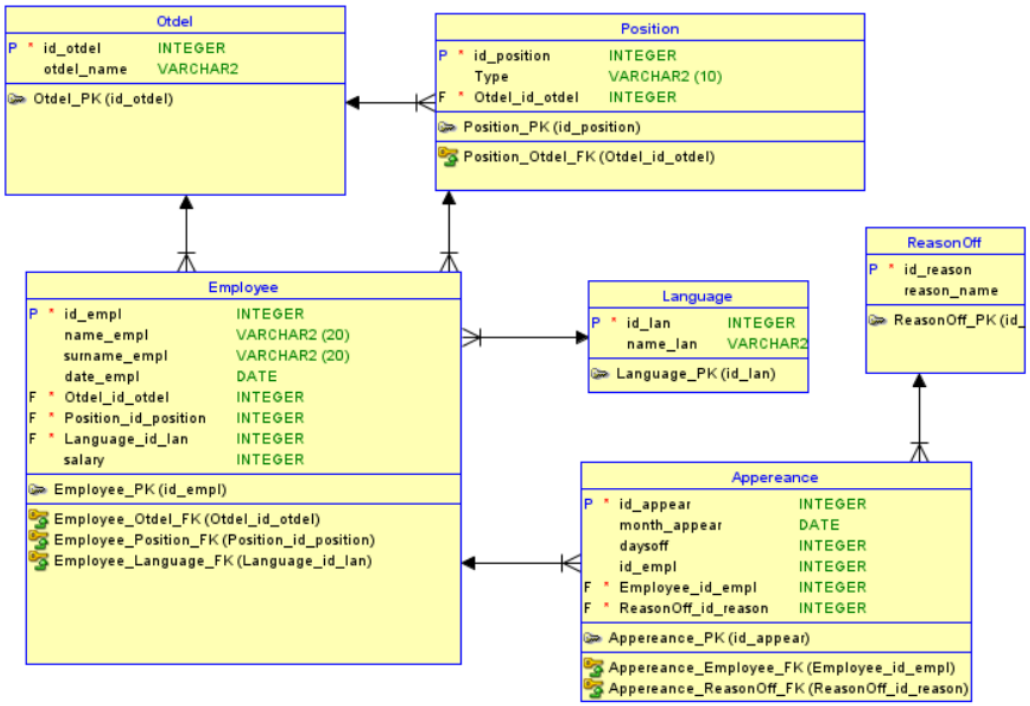
2. Цел на проекта

Създаване на релационна база данни, която позволява:

- Съхранение на служителска информация
- Управление на отдели и длъжности
- Отчитане на календарни месеци и работни дни
- Регистриране на отсъствия и причините за тях
- Проверка за коректност чрез тригери
- Генериране на справки чрез процедури и SELECT заявки

---

3. Структура на базата



Основни таблици:

- `EMPLOYEE`: информация за служителите
- `POSITION`: длъжности и заплати
- `OTDEL`: отдели
- `CALENDAR`: календар по месеци
- `REASONOFF`: причини за отсъствия
- `PRESENT_ABSENT`: отсъствия на служителите
- `PRESENT_ABSENT_REASON`: причини за конкретните отсъствия
- `LANGUAGE` и `EMPLOYEE_LANGUAGE`: владеене на езици

1. Създаване на основни таблици

```
CREATE TABLE Employee (
  id_empl NUMBER PRIMARY KEY,
  name_empl VARCHAR2(50),
  surname_empl VARCHAR2(50),
  hiredate_empl DATE,
  otdel_id_otdel NUMBER,
  position_id_position NUMBER
);
```

	ID_EMPL	NAME_EMPL	SURNAME_EMPL	HIREDATE_EMPL	OTDEL_ID_OTDEL	POSITION_ID_POSITION
1	1	Ivan	Georgiev	10-JAN-15	1	1
2	2	Maria	Ivanova	15-FEB-18	2	2
3	4	Elena	Stancheva	01-MAR-24	1	2
4	3	Petar	Dimitrov	20-MAY-22	3	3

```
CREATE TABLE Present_Absent (
  id_absence NUMBER PRIMARY KEY,
  daysabsent NUMBER,
  dayspresent NUMBER,
  employee_id_empl NUMBER,
  calendar_calendar_id NUMBER
);
```

	ID_ABSENCE	DAYSABSENT	DAYSPRESENT	EMPLOYE...	CALENDA...
1	7	5	25	1	1
2	1	2	28	1	1
3	2	5	25	2	1
4	3	0	30	3	2
5	4	5	23	1	2
6	5	4	24	2	3
7	6	6	22	3	4

```
CREATE TABLE Position (
    id_position NUMBER PRIMARY KEY,
    type VARCHAR2(50),
    salary NUMBER
);
```

	ID_POSITION	TYPE	SALARY
1	1	Developer	2700
2	2	HR Manager	2000
3	3	Accountant	2200
4	4	QA	2000

```
CREATE TABLE Otdel (
    id_otdel NUMBER PRIMARY KEY,
    otdel_name VARCHAR2(50)
);
```

	ID_OTDEL	OTDEL_NAME
1	1	IT
2	2	HR
3	3	Finance

```
CREATE TABLE Calendar (
    calendar_id NUMBER PRIMARY KEY,
    month DATE,
    year DATE,
    max_days NUMBER
);
```

	CALENDAR_ID	MONTH	YEAR	MAX_DAYS
1	1	01-JAN-24	01-MAR-24	31
2	2	01-FEB-24	01-MAR-24	29
3	3	01-MAR-24	01-MAR-24	31
4	4	01-APR-24	01-APR-24	30

```
CREATE TABLE ReasonOFF (
    reason_id NUMBER PRIMARY KEY,
    type VARCHAR2(50)
);
```

	REASON_ID	TYPE
1	1	Sick Leave
2	2	Vacation
3	3	Personal Day

```
CREATE TABLE Present_Absent_Reason (
    present_absent_id_absence NUMBER,
    reasonoff_reason_id NUMBER,
    number_of_days NUMBER
);
```

	REASONOFF_REASON_ID	PRESENT_ABSENT_ID_ABSENCE	NUMBER_OF_DAYS
1	3	4	0
2	2	7	3
3	2	6	4
4	1	6	2
5	1	1	2
6	2	2	2
7	3	2	3
8	1	4	3
9	2	4	2
10	2	5	2
11	3	5	2
12	1	5	0

```
CREATE TABLE Language (
    id_language NUMBER PRIMARY KEY,
    name_language VARCHAR2(50)
);
```

	ID_LANGUAGE	NAME_LANGUAGE
1	1	English
2	2	German
3	3	French

```
CREATE TABLE Employee_Language (
    id_empl NUMBER,
    id_language NUMBER
);
```

	ID_EMPL	ID_LANGUAGE
1	1	2
2	2	2
3	3	1
4	1	1
5	2	1

4. DML (Data Manipulation Language)

Вмъкване на данни (INSERT)

Таблица Employee:

```
INSERT INTO Employee (id_empl, name_empl, surname_empl, hiredate_empl, otdel_id_otdel, position_id_position)
VALUES (1, 'Ivan', 'Georgiev', TO_DATE('10-JAN-2015', 'DD-MON-YYYY'), 1, 1);

INSERT INTO Employee (id_empl, name_empl, surname_empl, hiredate_empl, otdel_id_otdel, position_id_position)
VALUES (2, 'Maria', 'Ivanova', TO_DATE('15-FEB-2018', 'DD-MON-YYYY'), 2, 2);
```

Таблица Calendar:

```
INSERT INTO Calendar (calendar_id, month, year, max_days)
VALUES (1, TO_DATE('01-01-2024', 'DD-MM-YYYY'), TO_DATE('2024', 'YYYY'), 31);
```

Таблица Otdel:

```
INSERT INTO Otdel (id_otdel, otdel_name) VALUES (1, 'IT');
```

Таблица Present_Absent:

```
INSERT INTO Present_Absent (id_absence, DaysAbsent, DaysPresent, Employee_id_empl, Calendar_calendar_id)
VALUES (1, 2, 28, 1, 1);
```

Таблица ReasonOFF:

```
INSERT INTO ReasonOFF (reason_id, type) VALUES (1, 'Sick Leave');
```

Таблица Present_Absent_Reason:

```
INSERT INTO Present_Absent_Reason (present_absent_id_absence, reasonoff_reason_id, number_of_days)
VALUES (1, 1, 2);
```

Актуализация на данни (UPDATE)

Промяна на заплата:

```
UPDATE Position
SET salary = 2700
WHERE type = 'Developer';
```

Промяна на фамилия:

```
UPDATE Employee
SET surname_empl = 'Georgiev'
WHERE name_empl = 'Ivan' AND surname_empl = 'Petrov';
```

Промяна на брой дни по причина:

```
UPDATE Present_Absent_Reason
SET number_of_days = 2
WHERE present_absent_id_absence = 2 AND reasonoff_reason_id = 2;
```

Изтриване на данни (DELETE)

Премахване на всички причини за дадено отсъствие:

```
DELETE FROM Present_Absent_Reason
WHERE present_absent_id_absence = 3;
```

Изтриване по конкретна причина:

```
DELETE FROM Present_Absent_Reason
WHERE present_absent_id_absence = 1
AND reasonoff_reason_id = 2;
```

---

5. Справки (SELECT заявки)

1. Служители с избрана фамилия

```
SELECT
    e.name_empl AS "Name",
    e.surname_empl AS "Surname",
    p.type AS "Position",
    o.otdel_name AS "Department",
    p.salary AS "Salary",
    e.hiredate_empl AS "Start date",
    l.name_language AS "Language"
FROM Employee e
JOIN Position p ON e.Position_id_position = p.id_position
JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
JOIN Employee_Language el ON e.id_empl = el.id_empl
JOIN Language l ON el.id_language = l.id_language
WHERE e.surname_empl = 'Ivanova'
ORDER BY e.surname_empl;
```

Name	Surname	Position	Department	Salary	Start dat	Language
Maria	Ivanova	HR Manager	HR	2000	15-FEB-18	English
Maria	Ivanova	HR Manager	HR	2000	15-FEB-18	German

---

2. Служители, които говорят английски

```
SELECT
    e.name_empl AS "Name",
    e.surname_empl AS "Surname",
    p.type AS "Position",
    o.otdel_name AS "Department",
    p.salary AS "Salary",
    e.hiredate_empl AS "Start date",
    l.name_language AS "Language"
FROM Employee e
JOIN Position p ON e.Position_id_position = p.id_position
JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
JOIN Employee_Language el ON e.id_empl = el.id_empl
JOIN Language l ON el.id_language = l.id_language
WHERE l.name_language = 'English'
ORDER BY e.surname_empl;
```

Name	Surname	Position	Department	Salary	Start dat	Language
Petar	Dimitrov	Accountant	Finance	2200	20-MAY-22	English
Ivan	Georgiev	Developer	IT	2700	10-JAN-15	English
Maria	Ivanova	HR Manager	HR	2000	15-FEB-18	English

---

3. Служители със заплата 2700

```
SELECT
    e.name_empl AS "Name",
    e.surname_empl AS "Surname",
    p.type AS "Position",
    o.otdel_name AS "Department",
    p.salary AS "Salary",
    e.hiredate_empl AS "Start date",
    l.name_language AS "Language"
FROM Employee e
JOIN Position p ON e.Position_id_position = p.id_position
JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
JOIN Employee_Language el ON e.id_empl = el.id_empl
JOIN Language l ON el.id_language = l.id_language
WHERE p.salary = 2700
ORDER BY e.surname_empl;
```

Name	Surname	Position	Department	Salary	Start dat	Language
Ivan	Georgiev	Developer	IT	2700	10-JAN-15	German
Ivan	Georgiev	Developer	IT	2700	10-JAN-15	English

---

4. Служители от отдел „HR“

```
SELECT
    e.name_empl AS "Name",
    e.surname_empl AS "Surname",
    p.type AS "Position",
    o.otdel_name AS "Department",
    p.salary AS "Salary",
    e.hiredate_empl AS "Start date",
```

```
    l.name_language AS "Language"
FROM Employee e
JOIN Position p ON e.Position_id_position = p.id_position
JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
JOIN Employee_Language el ON e.id_empl = el.id_empl
JOIN Language l ON el.id_language = l.id_language
WHERE o.otdel_name = 'HR'
ORDER BY e.surname_empl;
```

Name	Surname	Position	Department	Salary	Start dat	Language
Maria	Ivanova	HR Manager	HR	2000	15-FEB-18	German
Maria	Ivanova	HR Manager	HR	2000	15-FEB-18	English

---

5. Последните трима назначени

```
SELECT * FROM (
    SELECT
        e.name_empl AS "Name",
        e.surname_empl AS "Surname",
        e.hiredate_empl AS "Hire Date",
        p.type AS "Position",
        o.otdel_name AS "Department"
    FROM Employee e
    JOIN Position p ON e.Position_id_position = p.id_position
    JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
    ORDER BY e.hiredate_empl DESC
)
WHERE ROWNUM <= 3;
```

Ime	Familia	Data_post	Dlujnost	Otdel
Elena	Stancheva	01-MAR-24	HR Manager	IT
Petar	Dimitrov	20-MAY-22	Accountant	Finance
Maria	Ivanova	15-FEB-18	HR Manager	HR

---

6. Служители с длъжност „HR Manager“

```
SELECT
    e.name_empl AS "Name",
    e.surname_empl AS "Surname",
    p.type AS "Position",
    o.otdel_name AS "Department"
FROM Employee e
JOIN Position p ON e.Position_id_position = p.id_position
JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
WHERE p.type = 'HR Manager'
ORDER BY p.type, o.otdel_name;
```

Name	Surname	Position	Department
Maria	Ivanova	HR Manager	HR
Elena	Stancheva	HR Manager	IT

7. Служители, назначени в определен период

```
SELECT
    e.name_empl AS "Name",
    e.surname_empl AS "Surname",
    e.hiredate_empl AS "Start date"
FROM Employee e
WHERE e.hiredate_empl BETWEEN TO_DATE('01-JAN-2015', 'DD-MON-YYYY') AND
TO_DATE('01-JAN-2025', 'DD-MON-YYYY')
ORDER BY e.hiredate_empl;
```

Name	Surname	Start dat
Ivan	Georgiev	10-JAN-15
Maria	Ivanova	15-FEB-18
Petar	Dimitrov	20-MAY-22
Elena	Stancheva	01-MAR-24

8. Отсъствия по отдели

```
SELECT
    e.name_empl AS "Name",
    e.surname_empl AS "Surname",
    o.otdel_name AS "Department",
    p.type AS "Position",
    c.month AS "Month",
    r.type AS "Reason",
    par.number_of_days AS "Days for this reason"
FROM Present_Absent pa
JOIN Employee e ON pa.employee_id_empl = e.id_empl
JOIN Calendar c ON pa.calendar_calendar_id = c.calendar_id
JOIN Position p ON e.Position_id_position = p.id_position
JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
JOIN Present_Absent_Reason par ON pa.id_absence =
par.present_absent_id_absence
JOIN ReasonOFF r ON par.reasonoff_reason_id = r.reason_id
WHERE o.otdel_name = 'IT'
ORDER BY c.month, o.otdel_name, p.type;
```

Name	Surname	Department	Position	Month	Reason	Days for this reason
Ivan	Georgiev	IT	Developer	01-JAN-24	Sick Leave	2
Ivan	Georgiev	IT	Developer	01-FEB-24	Vacation	2
Ivan	Georgiev	IT	Developer	01-FEB-24	Sick Leave	3

6. Процедури (PL/SQL)

- `findBySurname` – намира служители по фамилия
- `last3Hired` – последните трима назначени

- `findByPosition` – по длъжност
- `findByPeriod` – по период на назначаване
- `absencesByDepartment` – справка за отсъствията по отдели

1. `findBySurname`

```
CREATE OR REPLACE PROCEDURE findBySurname(v_surname VARCHAR2)
AS
  CURSOR c_emp IS
    SELECT e.name_empl, e.surname_empl, p.type, o.otdel_name, e.hiredate_empl
    FROM Employee e
    JOIN Position p ON e.position_id_position = p.id_position
    JOIN Otdel o ON e.otdel_id_otdel = o.id_otdel
    WHERE e.surname_empl = v_surname;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Employee surname ' || v_surname);

  FOR record IN c_emp LOOP
    DBMS_OUTPUT.PUT_LINE(record.name_empl || ' ' || record.surname_empl || '
- ' ||
                         record.type || ' @ ' || record.otdel_name || ', ' ||
record.hiredate_empl);
  END LOOP;
END;
/
```

Процедурата намира и извежда всички служители с дадена фамилия.

```
Employee surname Ivanova
Maria Ivanova - HR Manager @ HR, 15-FEB-18
```

---

2. `last3Hired`

```
CREATE OR REPLACE PROCEDURE last3Hired
AS
  CURSOR c_emp IS
    SELECT name_empl, surname_empl, hiredate_empl
    FROM (
      SELECT * FROM Employee ORDER BY hiredate_empl DESC
    )
    WHERE ROWNUM <= 3;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Last 3 assigned:');

  FOR record IN c_emp LOOP
    DBMS_OUTPUT.PUT_LINE(record.name_empl || ' ' || record.surname_empl || '
- ' || record.hiredate_empl);
  END LOOP;
END;
/
```

Извежда последните трима служители, назначени в компанията.

```
Last 3 assigned:
Elena Stancheva - 01-MAR-24
Petar Dimitrov - 20-MAY-22
Maria Ivanova - 15-FEB-18
```

---

3. findByPosition

```
CREATE OR REPLACE PROCEDURE findByPosition(v_position VARCHAR2)
AS
  CURSOR c_emp IS
    SELECT e.name_empl, e.surname_empl, p.type, o.otdel_name
    FROM Employee e
    JOIN Position p ON e.position_id_position = p.id_position
    JOIN Otdel o ON e.otdel_id_otdel = o.id_otdel
    WHERE p.type LIKE '%' || v_position || '%';
BEGIN
  DBMS_OUTPUT.PUT_LINE('Employees with this position: ' || v_position);

  FOR record IN c_emp LOOP
    DBMS_OUTPUT.PUT_LINE(record.name_empl || ' ' || record.surname_empl || '
- ' ||
                         record.type || ', ' || record.otdel_name);
  END LOOP;
END;
/
```

Търси служители по наименование на длъжност.

```
Employees with this position: Manager
Maria Ivanova - HR Manager, HR
Elena Stancheva - HR Manager, IT
```

---

4. findByPeriod

```
CREATE OR REPLACE PROCEDURE findByPeriod(v_from DATE, v_to DATE)
AS
  CURSOR c_emp IS
    SELECT name_empl, surname_empl, hiredate_empl
    FROM Employee
    WHERE hiredate_empl BETWEEN v_from AND v_to;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Employees assigned in this period: ' || v_from || ' -
' || v_to);

  FOR record IN c_emp LOOP
    DBMS_OUTPUT.PUT_LINE(record.name_empl || ' ' || record.surname_empl || '
- ' || record.hiredate_empl);
```

```
  END LOOP;
END;
/
```

Показва всички служители, постъпили на работа в зададен период.

```
Employees assigned in this period: 01-JAN-15 ? 01-JAN-25
Ivan Georgiev - 10-JAN-15
Maria Ivanova - 15-FEB-18
Elena Stancheva - 01-MAR-24
Petar Dimitrov - 20-MAY-22
```

---

5. absencesByDepartment

```
CREATE OR REPLACE PROCEDURE absencesByDepartment(v_department VARCHAR2)
AS
  CURSOR c_abs IS
    SELECT e.name_empl, e.surname_empl, o.otdel_name, p.type,
           c.month, r.type AS reason, par.number_of_days
    FROM Present_Absent pa
    JOIN Employee e ON pa.employee_id_empl = e.id_empl
    JOIN Calendar c ON pa.calendar_calendar_id = c.calendar_id
    JOIN Position p ON e.Position_id_position = p.id_position
    JOIN Otdel o ON e.Otdel_id_otdel = o.id_otdel
    JOIN Present_Absent_Reason par ON pa.id_absence =
par.present_absent_id_absence
    JOIN ReasonOFF r ON par.reasonoff_reason_id = r.reason_id
    WHERE o.otdel_name = v_department;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Absences of department: ' || v_department);

  FOR record IN c_abs LOOP
    DBMS_OUTPUT.PUT_LINE(record.name_empl || ' ' || record.surname_empl ||
                         ' - ' || record.type || ' (' || record.reason || '):
' || record.number_of_days || ' days');
  END LOOP;
END;
/
```

```
Absences of a depratment: IT
Ivan Georgiev - Developer (Sick Leave): 2
Ivan Georgiev - Developer (Sick Leave): 3
Ivan Georgiev - Developer (Vacation): 2 ??
```

7. Автоматично генериране на ID при INSERT с помощта на SEQUENCE и TRIGGER

В базата данни, всяка таблица, която използва уникално ID (напр. `id_empl` или `id_otdel`), може да бъде настроена така, че това ID **да се генерира автоматично**, без да се въвежда ръчно.

Това става чрез:

- **SEQUENCE** – обект, който генерира последователни стойности (1, 2, 3, 4...)
- **BEFORE INSERT TRIGGER** – тригер, който взима следващата стойност от SEQUENCE и я поставя в полето преди записът да бъде добавен в таблицата

Пример 1: Автоматично ID за служител (`Employee`)

```
CREATE SEQUENCE emp_seq
START WITH 4
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

Това създава последователност (`SEQUENCE`) с име `emp_seq`, която ще започне от 4 и ще се увеличава с 1 при всяко извикване.
`NOCACHE` означава, че стойностите не се кешират в паметта, а `NOCYCLE` – че когато се стигне до краен лимит, няма да се повтарят.

```
CREATE OR REPLACE TRIGGER trg_employee_id
BEFORE INSERT ON Employee
FOR EACH ROW
BEGIN
  :NEW.id_empl := emp_seq.NEXTVAL;
END;
/
```

Този тригер се активира преди всяко ново вмъкване (`BEFORE INSERT`) в таблицата `Employee`.
Чрез `:NEW.id_empl := emp_seq.NEXTVAL;` той взима следващата стойност от `emp_seq` и я записва в полето `id_empl`.

Вмъкване на нов служител (без ID):

```
INSERT INTO Employee (name_empl, surname_empl, hiredate_empl, otdel_id_otdel,
position_id_position)
VALUES ('Elena', 'Stancheva', TO_DATE('01-MAR-2024', 'DD-MON-YYYY'), 1, 2);
```

Тук не въвеждаме ръчно ID — тригерът автоматично ще го попълни с `4`, `5`, `6` и т.н. според реда.

Пример 2: Автоматично ID за отдел (`Otdel`)

```
CREATE SEQUENCE otdel_seq
START WITH 4
INCREMENT BY 1;
```

Подобно на `emp_seq`, този SEQUENCE генерира ID-та за отдели, започвайки от 4.

```
CREATE OR REPLACE TRIGGER trg_otdel_id
BEFORE INSERT ON Otdel
FOR EACH ROW
BEGIN
  :NEW.id_otdel := otdel_seq.NEXTVAL;
END;
/
```

Тригерът автоматично записва стойност в `id_otdel`, без да се въвежда ръчно.

Примерен запис:

```
INSERT INTO Otdel (otdel_name)
VALUES ('Customer Support');
```

ID ще бъде зададен автоматично чрез `otdel_seq`.

8. Тригери

`trg_check_reason_days_total`

- Проверява дали общият брой дни по всички причини **не надвишава** зададените `DAYSABSENT`
- **BEFORE INSERT OR UPDATE**
- Извежда грешка, ако се нарушава логиката

```sql
CREATE OR REPLACE TRIGGER trg_check_reason_days_total
BEFORE INSERT OR UPDATE ON Present_Absent_Reason
FOR EACH ROW
DECLARE
  v_total_except_current NUMBER := 0;
  v_allowed_days NUMBER := 0;
BEGIN
  -- Сумираме всички причини за същото отсъствие, БЕЗ текущия запис (ако е UPDATE)
  SELECT NVL(SUM(number_of_days), 0)
  INTO v_total_except_current
  FROM Present_Absent_Reason
  WHERE present_absent_id_absence = :NEW.present_absent_id_absence
    AND (:NEW.reasonoff_reason_id IS NULL OR ROWID != :NEW.ROWID);

  -- Вземаме колко дни отсъствие са разрешени
  SELECT daysabsent
  INTO v_allowed_days
  FROM Present_Absent
  WHERE id_absence = :NEW.present_absent_id_absence;

  -- Сравняваме общата стойност + новите дни със зададените
  IF (v_total_except_current + :NEW.number_of_days) > v_allowed_days THEN
    RAISE_APPLICATION_ERROR(-20031, 'Total days for this absence ID exceed the allowed value!');
  END IF;
END;
/
```

Този тригер проверява дали общият брой дни по причини не надвишава стойността в полето DAYSABSENT за съответното отсъствие.
Ако се въведе твърде много – се генерира грешка, която спира операцията.

trg_log_absence_reason

- **AFTER INSERT**
- Извежда съобщение с информация за нововъведената причина за отсъствие

```
CREATE OR REPLACE TRIGGER trg_log_absence_reason
AFTER INSERT ON Present_Absent_Reason
FOR EACH ROW
DECLARE
  v_reason_type VARCHAR2(50);
BEGIN
  -- Get the reason type from ReasonOFF
  SELECT type
  INTO v_reason_type
  FROM ReasonOFF
  WHERE reason_id = :NEW.reasonoff_reason_id;

  -- Output log message
  DBMS_OUTPUT.PUT_LINE('Inserted absence reason: "' || v_reason_type ||
                       '" for absence ID: ' || :NEW.present_absent_id_absence ||
                       ' | Days: ' || :NEW.number_of_days);
END;
/
```

След успешно добавяне на причина, тригерът извежда съобщение чрез DBMS_OUTPUT, което съдържа типа на причината, ID на отсъствието и броя дни.

Пример: Добавяне на причина за отсъствие – лог изход от AFTER тригера

```
INSERT INTO Present_Absent_Reason (
    present_absent_id_absence,
    reasonoff_reason_id,
    number_of_days
) VALUES ( 7, 2, 3);
```

Обяснение:

Тази заявка добавя нов запис в таблицата `Present_Absent_Reason`, където:

- `present_absent_id_absence = 7` → това е ID на конкретно отсъствие, предварително създадено в таблицата `Present_Absent`
- `reasonoff_reason_id = 2` → отговаря на причината **"Vacation"**, дефинирана в таблицата `ReasonOFF`
- `number_of_days = 3` → задава брой дни за тази причина

Резултат от AFTER тригера:

След като записът е успешно въведен, тригерът `trg_log_absence_reason` извежда следното съобщение чрез `DBMS_OUTPUT`:

```
Inserted absence reason: "Vacation" for absence ID: 7 | Days: 3
```

Това служи като потвърждение, че записът е регистриран успешно, с ясна информация за вида и продължителността на отсъствието.

```
Inserted absence reason: "Vacation" for absence ID: 7 | Days: 3
```

Пример: Грешка при превишаване на разрешените дни – BEFORE тригер

```
INSERT INTO Present_Absent_Reason (
    reasonoff_reason_id,
    present_absent_id_absence,
    number_of_days
) VALUES (
    1, 2, 2
);
```

Обяснение:

Тази заявка се опитва да добави 2 дни по причина `"Sick Leave"` към отсъствие с `ID = 2`. Обаче, за `present_absent_id_absence = 2`, **вече е достигнат лимитът, определен от колоната `DAYSABSENT`** в таблицата `Present_Absent`.

BEFORE тригерът `trg_check_reason_days_total` засича, че:

```
всички дни до момента + новите 2 > разрешените (daysabsent)
```

И съответно **предотвратява операцията**, като извежда следната грешка:

Изход от тригера (в `Script Output`):

```
ORA-20031: Total days for this absence ID exceed the allowed value!
ORA-06512: at "VIKI.TRG_CHECK_REASON_DAYS_TOTAL", line 19
ORA-04088: error during execution of trigger
'VIKI.TRG_CHECK_REASON_DAYS_TOTAL'
```

Това показва, че бизнес правилото за максимален брой допустими отсъствия е наложено успешно чрез тригер.

```
Error starting at line : 317 in command -
INSERT INTO PRESENT_ABSENT_REASON (REASONOFF_REASON_ID, PRESENT_ABSENT_ID_ABSENCE, NUMBER_OF_DAYS)
VALUES (1, 2, 2)
Error report -
ORA-20031: Total days for this absence ID exceed the allowed value!
ORA-06512: at "VIKI.TRG_CHECK_REASON_DAYS_TOTAL", line 19
ORA-04088: error during execution of trigger 'VIKI.TRG_CHECK_REASON_DAYS_TOTAL'
```

trg_log_absence_reason

- **AFTER INSERT**
- Извежда съобщение с информация за нововъведената причина за отсъствие

```
CREATE OR REPLACE TRIGGER trg_log_absence_reason
AFTER INSERT ON Present_Absent_Reason
FOR EACH ROW
DECLARE
  v_reason_type VARCHAR2(50);
BEGIN
  -- Get the reason type from ReasonOFF
  SELECT type
  INTO v_reason_type
  FROM ReasonOFF
  WHERE reason_id = :NEW.reasonoff_reason_id;

  -- Output log message
  DBMS_OUTPUT.PUT_LINE('Inserted absence reason: "' || v_reason_type ||
                       '" for absence ID: ' || :NEW.present_absent_id_absence ||
                       ' | Days: ' || :NEW.number_of_days);
END;
/
```

След успешно добавяне на причина, тригерът извежда съобщение чрез DBMS_OUTPUT, което съдържа типа на причината, ID на отсъствието и броя дни.

---

10. Заключение

Проектът изпълнява зададените изисквания, демонстрира:

- Коректност на данните чрез тригери
- Използване на процедурно програмиране (PL/SQL)
- Богата и нормализирана структура
- Възможност за лесна справка и разширение