**ANKARA UNIVERSITY**

**COMPUTER ENGINEERING DEPARTMENT**

**COM258 LAB2**

**2020 SPRING**

The CHECKS table:

|  |  |  |  |
| --- | --- | --- | --- |
| **CHECK\_ID** | **PAYEE** | **AMOUNT** | **REMARKS** |
| 1 | Migros | 270.12 | Groceries |
| 2 | Petrol Ofisi | 105.00 | Gas (For Car) |
| 3 | TCDD | 25.00 | Train to Konya |
| 4 | Turkcell | 528.00 | Cellular Phone |
| 5 | Baskent Elektrik | 113.78 | Electricity Bill |
| 6 | Flo | 175.00 | Shoes |
| 7 | Migros | 315.90 | Groceries |
| 8 | Allianz Sigorta | 365.43 | Car Insurance |

**SELECT STATEMENT**

If we use SELECT \* FROM CHECKS; we get all columns of data.

We do not have to follow predefined order; we can also change the order.

SELECT PAYEE, REMARKS, AMOUNT, CHECK\_ID FROM CHECKS;

Also we can select individual columns:

SELECT PAYEE, REMARKS FROM CHECKS;

In that example, we can observe the repeated data;

If we use SELECT DISTINCT PAYEE, REMARKS FROM CHECKS; only one instance of the duplicated data is shown.

**CONDITIONS**

If we want to get specific records, we should use expressions; to define an expression, we use WHERE clause;

SELECT AMOUNT, CHECK\_ID FROM CHECKS WHERE PAYEE = ‘TCDD’;

If we need one more condition, we may put additional conditions into the WHERE clause.

SELECT REMARKS FROM CHECKS WHERE PAYEE = ‘TCDD’ and AMOUNT>10;

**OPERATORS**

The arithmetic operators are plus (+), minus (-), divide (/), multiply (\*) and modulo (%)

Try these queries:

* SELECT AMOUNT,AMOUNT+0.15 FROM CHECKS;
* SELECT PAYEE, -AMOUNT AS NEWAMOUNT FROM CHECKS;
* SELECT PAYEE, (AMOUNT/2) AS NEWAMOUNT FROM CHECKS;
* SELECT CHECK\_ID, (AMOUNT\*0.9) AS NEWAMOUNT FROM CHECKS;

**COMPARISON OPERATORS**

Comparison operator compares expressions and returns one of the three values: TRUE, FALSE, Unknown.

* SELECT \*

FROM CHECKS

WHERE PAYEE=’Migros’;

For comparison, we can also use greater than (>), less than (<), greater than or equal(>=), less than or equal(<=) and inequalities(!=)

* SELECT \*

FROM CHECKS

WHERE PAYEE !=’Petrol Ofisi’;

If you want to select parts of a database that fit a pattern but were not quite exact match, you should use LIKE

* SELECT \*

FROM CHECKS

WHERE REMARKS LIKE ‘%EL%’;

%EL, get occurrence that ended with EL

EL%, get occurrence that started with EL

**LOGICAL OPERATORS**

* SELECT PAYEE

FROM CHECKS

WHERE AMOUNT <=150

**AND**

CHECK\_ID >3;

**IN and BETWEEN**

You can use OR to define multiple conditions. For example:

SELECT \* FROM FRIENDS WHERE STATE=’CA’ OR STATE=’CO’ OR STATE=’LA’;

But you can use IN without using multiple OR:

SELECT \* FROM FRIENDS WHERE STATE IN (‘CA’,’CO’,’LA’);

Also you can use BETWEEN to define a range

SELECT \* FROM CHECKS WHERE AMOUNT >25 AND AMOUNT<200;

with using BETWEEN instead of AND

SELECT \* FROM CHECKS WHERE AMOUNT BETWEEN 25 AND 200;

**ORDER BY**

SELECT \* FROM CHECKS ORDER BY CHECK\_ID DESC;

This query returns all the checks with decreasing order of CHECK\_ID.

**LAB EXERCISES**

Use the **CHECKS** table to answer the following questions.

|  |  |  |  |
| --- | --- | --- | --- |
| **CHECK\_ID** | **PAYEE** | **AMOUNT** | **REMARKS** |
| 1 | Migros | 270.12 | Groceries |
| 2 | Petrol Ofisi | 105.00 | Gas (For Car) |
| 3 | TCDD | 25.00 | Train to Konya |
| 4 | Turkcell | 528.00 | Cellular Phone |
| 5 | Baskent Elektrik | 113.78 | Electricity Bill |
| 6 | Flo | 175.00 | Shoes |
| 7 | Migros | 315.90 | Groceries |
| 8 | Allianz Sigorta | 365.43 | Car Insurance |

**1.** Write a query that returns all checks (CHECK\_ID, PAYEE, and AMOUNT) in the database in which PAYEE begins with M or P.

**2.** Write a query that returns all checks (CHECK\_ID, PAYEE, and AMOUNT) that is related to Car expenses.

**3.** Write a query that returns all PAYEEs in which the check AMOUNT is greater than 200.00. Only one instance of duplicate data is shown.

**4.** Write a query that returns all the checks with the decreasing order of the check AMOUNT.

**5.** Write a query that returns all the checks (CHECK\_ID, PAYEE, and AMOUNT) that is related to Shoes or Groceries expenses.