

Geral

- Entrar no mysql
→ mysql -u (nome de utilizador) -p OU → mysql -p
- Sair do mysql: quit

SQL

- Criar BD: CREATE DATABASE employees;
Apagar BD/tabela: DROP DATABASE/TABLE employees (IF EXISTS);
- Usar BD: USE employees;
- Confirmar BD a ser usada: SELECT DATABASE();
- Mostrar as BDs/tabelas: SHOW DATABASE; / SHOW TABLES;
- Carregar um ficheiro: SOURCE employees_schema.sql;
- Data e hora atual: SELECT NOW();
- Data atual: SELECT CURDATE();
- Informações sobre a tabela: DESCRIBE employee;
- SELECT * FROM employee WHERE job_name = 'SALESMAN' ORDER BY emp_name;
Mostrar todos os dados da tabela: SELECT * FROM employee;
Condição WHERE só mostra as pessoas com o emprego 'salesman'
ORDER BY ordena pela coluna escolhida (ASC ou DESC)
GROUP BY tem uma função parecida com order, mas só é usada para funções como o count
- Mudar o nome da coluna: SELECT emp_name (AS) nome FROM employee;
* AS é opcional
- Contar o número de elementos: COUNT(x)
Matemática: COUNT(), SUM(), AVG(), MAX(), MIN()
Aritméticas - abs(), ceil(), floor(), exp(), power(x,x), sqrt()
Igual: = / Diferente: <>
Exemplo do count:
SELECT job_name, count(job_name) FROM employee GROUP BY job_name;
- DIFERENÇA DE ANOS:
SELECT emp_name, hire_date, **TIMESTAMPDIFF**(YEAR, hire_date, CURDATE())
Experience FROM employee WHERE **TIMESTAMPDIFF**(YEAR, hire_date, CURDATE()) > 17;
- USO DO INNER JOIN:
→ SELECT employee.emp_name, department.dep_name FROM employee **INNER JOIN** department WHERE employee.dep_id = department.dep_id ORDER BY department.dep_name, employee.emp_name;
→ SELECT employee.emp_name, department.dep_location FROM employee **INNER JOIN** department **ON** employee.dep_id = department.dep_id WHERE (department.dep_location = 'SYDNEY' OR department.dep_location = 'MELBOURNE') AND (salary BETWEEN 2000 AND 5000) AND YEAR(hire_date) = 2001 ORDER BY employee.emp_name;

- **SELECT DENTRO DE SELECT:**
 Compara a hire_date de todos com a da Adelyn
 → `SELECT * FROM employee WHERE hire_date < (SELECT hire_date FROM employee WHERE emp_name = 'ADELYN');`
 Mostra todos os empregados que são managers de outros
 → `SELECT * FROM employee WHERE emp_id IN (SELECT manager_id FROM employee);`

LEIS DE MORGAN

- $(x.y)' = x'.y'$
 $(x+y)' = x'.y'$

TEOREMAS DE ÁLGEBRA DE BOOLE

- $X = 0$ se $X \neq 1$ e $X = 1$ se $X \neq 0$
- $X = 0$ se $X' = 1$ e $X = 1$ se $X' = 0$
- $0.0 = 0$ $1.1 = 1$ $0.1 = 1.0 = 0$ (E)
- $0+0 = 0$ $1+1 = 1$ $0+1 = 1+0 = 1$ (OU)
- $x + 0 = x$ $x.0 = 0$ $x+x = x$ $(x')' = x$
 $x + 1 = 1$ $x.1 = x$ $x.x = x$
 $x+x' = 1$ $x.x' = 0$
- $(x+y)+z = x+(y+z)$ $(x.y).z = x.(y.z)$
- $x.y+x.z = x.(y+z)$ $(x+y).(x+z) = x+(y.z)$
- $x+x.y = x$ $x.(x+y) = x$
- $x.y + x.y' = x$ $(x+y).(x+y') = x$
- $x.y + x'.z + y.z = x.y+x'.z$ $(x+y).(x'+z).(y+z) = (x+y).(x+z)$

Simplificações:

$$\begin{aligned}
 \rightarrow S &= A'.B'.C' + A'.B.C' + A.B'.C \Leftrightarrow S = A'.C'.B' + A'.C'.B + A.B'.C \Leftrightarrow \\
 S &= A'.C'.(B' + B) + A.B'.C \Leftrightarrow S = A'.C'.(1) + A.B'.C \Leftrightarrow \\
 \mathbf{S} &= \mathbf{A'.C' + A.B'.C} \\
 \rightarrow S &= A'.B'.C' + A'.B.C + A'.B.C' + A.B'.C' + A.B.C' \Leftrightarrow \\
 S &= A'.B'.C' + A'.B.C + A'.B.C' + A.B'.C' + A.B.C' \Leftrightarrow \\
 S &= A'.B.C + (A'.B' + A'.B + A.B' + A.B).C' \Leftrightarrow S = A'.B.C + (A'.B' + A'.B + A.B' + A.B).C' \Leftrightarrow \\
 S &= A'.B.C + (A'.(B' + B) + A.(B' + B)).C' \Leftrightarrow S = A'.B.C + (A'.(1) + A.(1)).C' \Leftrightarrow \\
 S &= A'.B.C + (A' + A).C' \Leftrightarrow S = A'.B.C + (1).C' \Leftrightarrow S = A'.B.C + C' \Leftrightarrow \\
 \mathbf{S} &= \mathbf{A'.B + C'}
 \end{aligned}$$