**Gabriel Araújo de Sousa – GU3027261**

**P1 – Banco de dados II**

1. A) Um gestor de vendas solicitou ao cientista de dados a média salarial por departamento da empresa em que trabalham e apresentou o gráfico abaixo. Gere o script SQL para criação do gráfico em uma única query:R:

select d.department\_name,avg(e.salary) as "Media\_salarial"

from employees e

inner join departments d

on d.department\_id = e.department\_id

group by d.department\_id

order by avg(e.salary);

B) O analista de dados, responsável por extrair informações do banco, resolveu tirar férias e sobrou para você realizar algumas extrações. Crie o script SQL para calcular a diferença salarial do funcionário com relação à média do departamento e ordene pelo nome do departamento:

R:

select ep.employee\_id,

ep.first\_name,

ep.last\_name,

ep.salary,(select avg(e.salary)

from employees e

inner join departments d

on d.department\_id = e.department\_id

where d.department\_id = ep.department\_id

group by d.department\_id) as media,

dp.department\_name,

(ep.salary - (select avg(e.salary)

from employees e

inner join departments d

on d.department\_id = e.department\_id

where d.department\_id = ep.department\_id

group by d.department\_id)) as diferenca

from employees ep

inner join departments dp

on dp.department\_id = ep.department\_id

order by dp.department\_name;

C) A quantidade de funcionários abaixo da média de salários por departamento pode ser visualizada na figura abaixo. Gerar o script SQL:

R:

SELECT d.department\_name,

COUNT(CASE WHEN e.salary < avg\_salaries.avg\_salary THEN 1 ELSE NULL END) AS salarios\_abaixo\_da\_media,

COUNT(\*) AS quantidade\_de\_funcionarios

FROM departments d

JOIN (

SELECT department\_id, AVG(salary) AS avg\_salary

FROM employees

GROUP BY department\_id

) AS avg\_salaries ON d.department\_id = avg\_salaries.department\_id

JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name;

1. Segundo a LGPD (lei geral de proteção de dados) <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm> Art. 5º Para os fins desta Lei, considera-se: II - dado pessoal sensível: dado pessoal sobre origem racial ou étnica, convicção religiosa, opinião política, filiação a sindicato ou a organização de caráter religioso, filosófico ou político, dado referente à saúde ou à vida sexual, dado genético ou biométrico, quando vinculado a uma pessoa natural; Com base no modelo lógico abaixo, proponha o cumprimento à LGPD, sem exclusão de informação:

R: Para que o modelo mencionado esteja de acordo com a LGPD, algumas ações podem ser tomadas, como: criptografar as informações sensíveis presentes na tabela paciente, como os campos DNA e Etnia, para garantir a confidencialidade e integridade dos dados. implementar controles de acesso aos dados sensíveis, limitando o acesso a profissionais autorizados e com justificativa adequada para acessá-los. E por último, adotar políticas de retenção de dados.

1. Uma consulta está realizando um FULL TABLE SCAN na tabela médico do banco de dados hospital. Essa consulta está demandando tempo desnecessário. Proponha uma otimização na consuta abaixo e explique os motivos. Utilize o modelo lógico da questão 2. Utilize o banco de dados do link: <https://github.com/renatobdo/BD2/tree/main/P1>

R: Para otimizar a consulta podemos criar um índice na coluna especialidade da tabela medico sendo assim, o índice é criado através do seguinte comando:

CREATE INDEX idx\_especialidade ON medico (especialidade);

1. A) Com base no modelo lógico abaixo crie uma trigger que será acionada quando um novo comentário é realizado na tabela comentário.

R: CREATE TRIGGER tr\_comentario AFTER INSERT ON comentario

FOR EACH ROW

BEGIN

DECLARE projeto\_id INT;

DECLARE usuario\_id INT;

SELECT id\_projeto, id\_usuario INTO projeto\_id, usuario\_id FROM comentario WHERE id = NEW.id;

INSERT INTO likes\_por\_projeto (id\_projeto, id\_usuario) VALUES (projeto\_id, usuario\_id);

END;

B) - criar uma trigger que dispara ao ser atingido 1000 likes por projeto.

R: CREATE TRIGGER likes\_por\_projeto\_trigger AFTER INSERT ON likes\_por\_projeto

FOR EACH ROW

BEGIN

DECLARE likes\_count INT;

SELECT COUNT(\*) INTO likes\_count FROM likes\_por\_projeto WHERE id\_projeto = NEW.id\_projeto;

IF likes\_count = 1000 THEN

INSERT INTO notificacao (mensagem) VALUES ('O projeto ' || (SELECT titulo FROM projeto WHERE id = NEW.id\_projeto) || ' atingiu 1000 likes!');

END IF;

END;

1. Crie uma trigger que será acionada quando é inserido um valor na coluna dataNascimento, cuja data de aniversário seja daqui a uma semana. Ela deverá mostrar a seguinte mensagem: “Olá, membros, o aniversário do membro\_fulano é daqui a 7 dias. Vamos comprar uma lembrança para ele?”

R:

DELIMITER $$

CREATE TRIGGER trigger\_aniversario\_proximo

AFTER INSERT ON membros

FOR EACH ROW

BEGIN

IF DATE\_ADD(NEW.dataNascimento, INTERVAL 7 DAY) = DATE\_ADD(CURDATE(), INTERVAL 7 DAY) THEN

INSERT INTO lembrete (membroId, mensagem) VALUES (NEW.id, CONCAT('Olá membros, o aniversário do membro\_', NEW.nome, ' é daqui a 7 dias. Vamos comprar uma lembrança para ele?'));

END IF;

END $$

DELIMITER ;

1. Crie uma trigger que irá atualizar o atributo preco\_desconto em 8% antes de inserir na tabela produto.

R:

CREATE TRIGGER atualiza\_preco\_desconto

BEFORE INSERT ON produto

FOR EACH ROW

SET NEW.preco\_desconto = NEW.preco\_normal \* 0.92;

1. Como o obter o resultado abaixo? Utilize o banco de dados hr no link: <https://github.com/renatobdo/BD2/blob/main/BancosDeDados/HR.sq>

R:

select d.department\_name, min(salary) as "min\_salary", max(salary) as "max\_salary", avg(salary) as "average\_salary"

from employees e

inner join departments d on d.department\_id = e.department\_id

group by e.department\_id

order by d.department\_name