OS BENEFÍCIOS DA UTILIZAÇÃO DE VIEWS EM BANCO DE DADOS RELACIONAIS

THE BENEFITS OF USING VIEWS IN A RELATIONAL DATABASE

Ana Beatriz Ferreira dos Santos
Graduando em Banco de Dados pela FATEC Bauru
E-mail: ana.santos118@fatec.sp.gov.br

Ricardo Ferreira dos Santos
Graduando em Banco de Dados pela FATEC Bauru
E-mail: ricardo100.eu@gmail.com

Luís Alexandre da Silva
Docente na FATEC Bauru
E-mail: luis.silva51@fatec.sp.gov.br

RESUMO. Para qualquer empresa que utiliza algum tipo de banco de dados relacional e que pretende fazer sempre uma consulta em alguma das tabelas criadas e não necessariamente precise revelar todo o conteúdo ou também quer facilitar no caso de ter que consultar informações de muitas tabelas, nesses casos é utilizado uma visão. Com isso esse artigo tem como objetivo apresentar uma ferramenta para facilitar e resolver esse problema. Essa ferramenta pode ser usada em qualquer tipo de banco seja ele centralizado ou distribuído desde que implemente a funcionalidade. Sãoesperadas com esse trabalho que seja apresentado quais as eficácias de uma view no banco de dados relacionais.

Palavras-chave: Banco de dados relacionais. View. View materializada.

ABSTRACT. *For any company that uses some type of relational database and that always wants to make a query in any of the created tables and does not necessarily need to reveal all the content or also wants to facilitate it in case of having to consult information from many tables, in these cases it is used a vision. This article aims to present a tool to facilitate and solve this problem. This tool can be used in any type of bank, whether centralized or distributed, as long as it implements the functionality. This work is expected to show the effectiveness of a view in the database.*

Keywords:*Relational database. View. View materialized.*

1. INTODUÇÃO

Antigamente a forma de armazenamento de dados nas empresas era por fichas de papel e organizadas em arquivos físicos em pastas. Além das dificuldades de extrair e manter organizado os dados, o acesso a essas informações dependia muito da localização geográfica dos arquivos, com isso os arquivos físicos evoluíram e passaram a seremdigitais, no caso, os bancos de dados relacionais.

Atualmente há muitos modelos de Sistema Gerenciado de Banco de Dados (SGBD) como, orientado a objetos, orientado a documentos etc.,porém o mais utilizado ainda é o banco de dados relacional.

Um dos motivos para se utilizar a view é que ela autoriza o usuário a consultar somente as informações necessárias, não permitindo o acesso de toda a tabela do banco de dados das empresas.

Em geral o trabalho tem como objetivo analisar o desempenho de viewsem banco de dados.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. Banco de Dados

Segundo Silva (2015),afirma que uma das definições de banco de dados (SGBD) se trata de uma coleção de informações que se relacionam de modo que criem algum sentido, isto é, uma estrutura bem organizada de dados que permite a extração de informações. Assim, são muito importantes para empresas tornarem-se a principal peça dos sistemas de informação.

Banco de dados é uma aplicação que possibilita receber e armazenar dados, está em todos os tipos de sistemas como computadores pessoais, servidores, smartphones, já que esses guardam informações necessárias para o funcionamento dos sistemas. Existem dois principais modelos de banco de dados, o relacional e o não relacional (NoSQL). O que faz com que o banco seja relacional ou não relacional é a forma a qual os dados são inseridos e organizados nele.

2.2. View

De acordo com Alves (2015), uma view é uma consulta armazenada no banco de dados, podendo ser consultada como se fosse uma tabela. Uma de suas principais funções é controlar a segurança do banco de dados, geralmente ela é criada com campos que determinado perfil de usuário pode acessar, concedendo acesso apenas a essa view e não a tabelas diretamente.

A view também é utilizada para apresentar informações mais organizadas para o usuário, ela já ficaria armazenada no próprio banco e o usuário não precisaria elabora uma consulta complexa.

Figura 1 – Exemplo de tabela em um banco de dados

FUNCIONARIO				
CPF	Nome	E-mail	Salário	ID_DEPARTAMENTO
123.456.789-10	João	joao@exemplo.com	R$ 4.500,00	1
111.222.333-44	Gustavo	gustavo@exemplo.com	R$ 2.500,00	1
222.333.444-55	Pedro	pedro@exemplo.com	R$ 3.500,00	2

Fonte:Alves (2015)

Figura 2 – Exemplo de tabela em um banco de dados

DEPARTAMENTO	
ID_DEPARTAMENTO	NOME
1	Financeiro
2	Recursos Humanos

Fonte:Alves (2015)

Considerando que um usuário precisa de uma lista atualizada dos funcionários e seus departamentos. Em questões de segurança, não pode ser fornecido mais nenhumas informações como CPF, e-mail e salário dos funcionários. A melhor forma é criar uma view onde essas informações não são apresentadas e fornece ao usuário acesso apenas a esta view, ou seja, o usuário só terá acesso ao nome e ao departamento.

Figura 3 – Exemplo de view em um banco de dados

V_FUNCIONARIO_DEPARTAMENTO	
FUNCIONARIO	DEPARTAMENTO
Gustavo	Financeiro
João	Financeiro
Pedro	Recursos Humanos

Fonte:Alves (2015)

A consulta dessa view poderia ser: SELECT F.NOME AS FUNCIONARIO, D.NOME AS DEPERTAMENTO FROM FUNCIONARIO F INNER JOIN DEPARTAMENTO D ON D.ID_DEPARTAMENTO = F.ID_DEPARTAMENTO

A view realiza uma consulta (query) em tempo de execução. Em uma view simples essa consulta que é armazenada. Essa consulta pode ter condições próprias para restringir os dados que serão visualizados pelos usuários, tanto horizontal (colunas que serão apresentadas) quanto vertical (linhas que serão apresentadas).

Se no mesmo caso acima, tirarmos um dos departamentos “Recursos Humanos”, por exemplo, bastaria colocar uma condição na cláusula “WHERE” da view (whereid_departamento<> 2).

2.3. View Materializadas

ParaAlves (2015), visão materializada (VM) é uma view comum, porém o que armazena não é a consulta e sim o resultado dela. Uma VM é uma tabela real no banco de dados que é atualizada sempre que ocorrer uma atualização em alguma tabela, usada pela sua consulta. Outra forma de atualização da view materializada é atualizá-la diariamente.

Views Materializadas podem ser extremamente interessantes de acordo com a necessidade de um negócio ou mesmo cenário que se faça necessário, devido a não necessitar que todos os pontos (nós) estejam disponíveis, por exemplo, quando se trata de uma visão que consome dados de vários bancos de dados que estiverem organizados de forma distribuída, dessa forma também agilizando muito a pesquisa dos dados, pois suprime a necessidade de serialização/desserialização dos dados entre os nós, suprime também a dependência dos recursos de comunicação de telefonia e redes (nesse caso, reduzindo muito o tempo de latência), além de trazer

os pontos importantes de uma view comum como: Segregação dos dados, segurança da informação, abstração do conhecimento da estrutura física dos dados entre outros.

Utilizando ainda o exemplo acima. Se a view V_FUNCIONARIO_DEPARTAMENTO for materializada e sua programação for efetuada diariamente, por exemplo, sempre que a tabela departamento ou funcionário receber uma inclusão, alteração ou exclusão, a consulta da view também será executada e o resultado será armazenado.

Embora a consulta na view fique mais rápida com o pré-procedimento da consulta interna, o processo de escrita no banco de dados fica mais lento, pois é necessário executar a consulta interna da view materializada sempre que algum dado sofrer atualização (caso seja essa a programação de atualização da referida view materializada).

3. BANCO DE DADOS RELACIONAL

Segundo Puga, França e Goya (2013), no ano de 1960,Dr. Edgar F. Codd,um pesquisador e cientista da IBM, pesquisava maneira de lidar com um volume grande de dados. Não feliz com modelos de banco de dados daquela época, teve a ideia de aplicar disciplinas e estruturas matemáticas na administração de dados, para resolver vários problemas encontrados nos outros, como: redundância de dados, dependência física na implementação e falhas na integridade de dados.

Em 1970, Dr.Codd apresentou e patenteou seu trabalho como “Um Modelo Relacional de Dados para Grandes Bancos de Dados Compartilhados”. Neste trabalho, apresentou o modelo relacional de banco de dados, se baseando em duas vertentes das exatas: na teoria dos conjuntos e na lógica de predicados de primeira ordem.

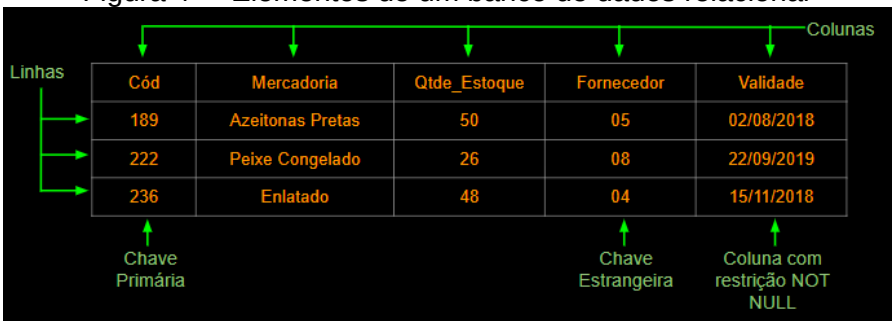
O nome Modelo Relacional vem do termo “relação”, que é uma parte da teoria dos conjuntos.

Uma das características principais do modelo é a possibilidade de relacionar várias tabelas, evitando a redundância no armazenamento de dados. O relacionamento nesse modelo pode ocorrer de três formas:

a) Um-para-um;
b) Um-para-muitos;
c) Muitos-para-muitos.

Com um exemplo de duas tabelas (A e B) estarão conectadas quando uma (A) tiver um campo de compartilhamento, que poderá ser utilizado em outra (B), assim diminuindo o excesso de dados nas inclusões de registros e facilitando próximas buscas e pesquisas.

Figura 4 – Elementos de um banco de dados relacional



Fonte:Reis (2019)

3.1.1 Vantagens das views em banco de dados relacional

As views são muito usadas para ajudar dar entendimento ao projeto lógico do banco de dados.

São muitos os motivos e vantagens para utiliza-las nos projetos.

a) É possível ter o controle das informações que o usuário terá acesso. Selecionando apenas os campos que deseja disponibilizar.

b) As consultas são simplificadas, não sendo necessário fornecer parâmetros sempre que as consultas forem executadas.

c) Há mais segurança, pois a view é um objeto do banco de dados, assim possível atribuírem permissões de usuários.

3.1.2 Desvantagens

a) A view acaba escondendo uma complexidade da query, podendo assim enganar o desenvolvedor quanto odesempenho necessário para acessar as informações.

b) Cria uma camada extra, com isso são mais objetos para administrar, considerando um aumento de complexidade.

c) Pode limitar até demais o acesso do usuário, impedindo certas tarefas.

3.1. Oracle

De acordo com Scudero (2016), originada nos anos 80 a Oracle foi criado por Larry Ellison, sendo hoje uma das maiores empresas de tecnologia, desde 2009 é proprietária das linhas de software Java.

Seu principal produto é o SGBD desde seu lançamento no mercado é aperfeiçoado e desenvolvido para atender as necessidades das empresas e do mundo. Há diversas versões do software e a cada uma delas tem características que a tornam ideal á diferentes modelos de negócio, é um software para as necessidades de empresas de médio e grande porte. Para manipulação e gestão do sistema é utilizada a linguagem PL/SQL, linguagem desenvolvida pela Oracle, a partir do SQL (ANSI), expandindo a sua capacidade original.

3.2. MYSQL

Para Pacievitch (2011), o Mysql foi criado na Suécia, por David Axmark, Allan Larsson e Michael Widenius. O Mysql é um SGBD, que usa a linguagem SQL como interface. Esse banco de dados é conhecido pela facilidade de uso, muitas empresas como NASA, HP, Bradesco, Sony e muitas outras utilizam. Alguns dos motivos para ser tão usado é sua interface simples e sua capacidade de rodar em vários sistemas operacionais. Mesmo sendo um dos mais utilizados esse banco está sempre em desenvolvimento, pois ainda são encontradas algumas falhas nele, que são resolvidos com atualizações.

O MySQL é protegido por uma licença de software livre, desenvolvida pela GNU, também um dos programas que normalmente vem instalado com o GNU/Linux. Banco muito utilizado para sites de cadastro de lojas.

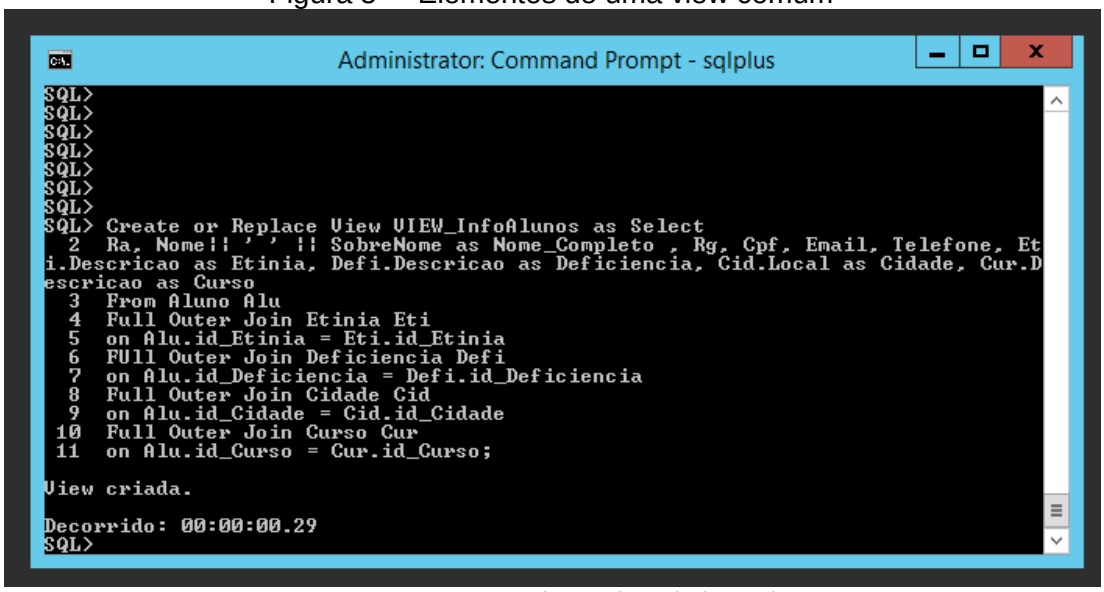
4. MATERIAIS E MÉTODOS

Para o desenvolvimento desse trabalho, foi utilizado um servidor de banco de dados Oracle e umMysql. Foi decidido utilizar esses bancos, pois são sistemas com alta qualidade e com melhor eficiência e alta compatibilidade entre eles, o que possibilita a criação de links homogêneos e heterogêneos.

Diante da ideia proposta,foram realizados testes comviews e views materializadas no banco de dados relacional para comprovar suas eficiências.

De acordo com a figura 5 é realizado a criação de uma viewdentro do banco de dado relacional Oracle, com informações de cadastro do aluno na faculdade, contendo Ra, Nome, Sobrenome, RG, CPF, etc.

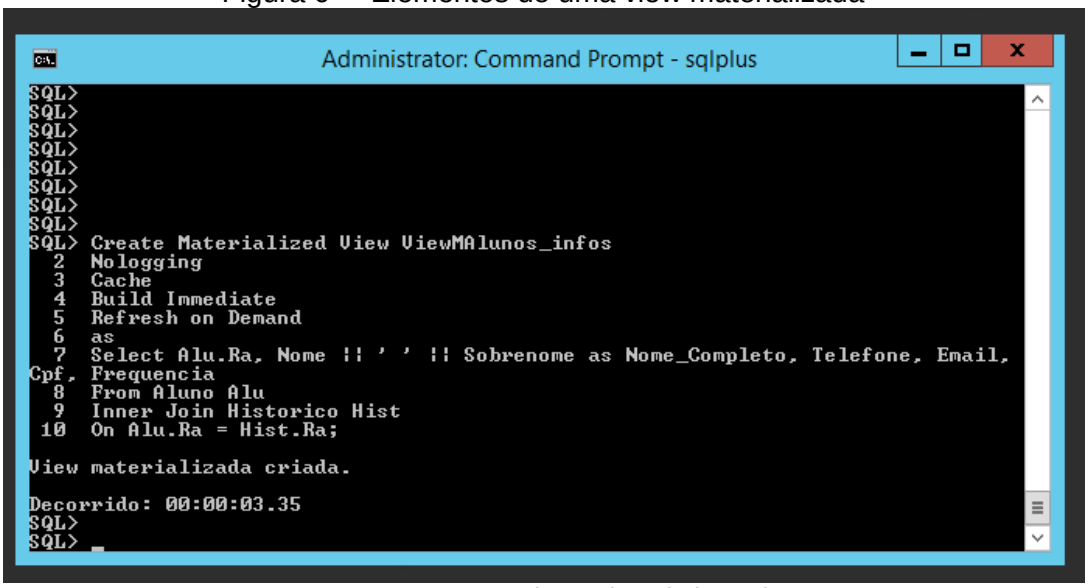
Figura 5 – Elementos de uma view comum



Fonte: Ana Beatriz e Ricardo(2020)

Na Figura 6 é uma view materializada sendo criada, podemos observar as mesmas informações da tabela acima, diferenciando o tempo ocorrido, uma com apenas 29 centésimo de segundos, e a outra com 3.35segundos.

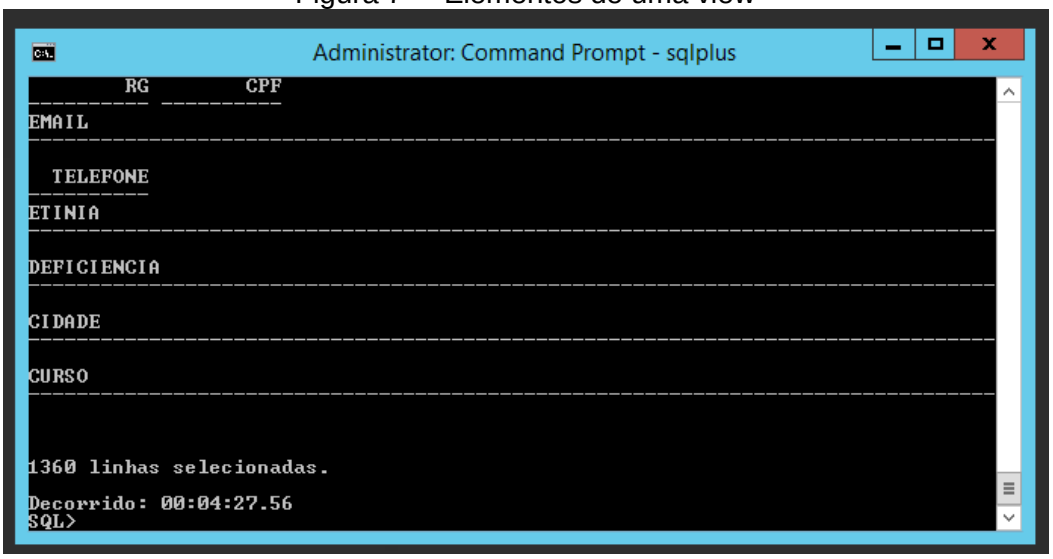
Figura 6 – Elementos de uma view materializada



Fonte: Ana Beatriz e Ricardo(2020)

Já na Figura 7 são as informações similares tirada da view de uma mesma coluna, da Figura 5 (view comum) sendo executada para mostras o resultado e o tempo de execução do select.

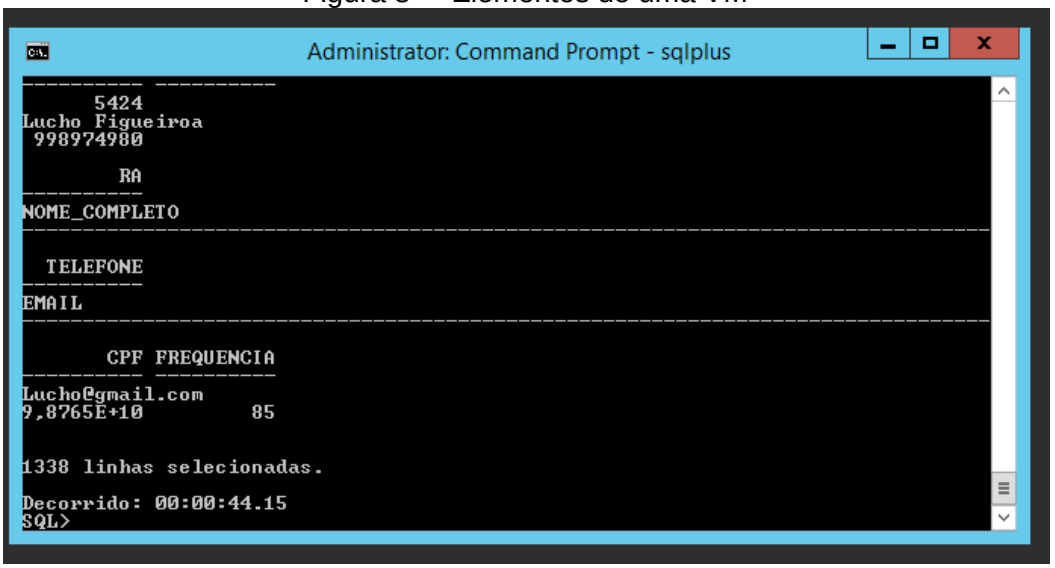
Figura 7 – Elementos de uma view



Fonte: Ana Beatriz e Ricardo(2020)

De acordocom a Figura 8 é a VM sendo executado com as mesmas informações da Figura 6, para mostrar seu tempo de execução. Assim tendo a view com o tempo de 4 minutos e 27.56 segundos e a view materializada com apenas 44.15 segundos

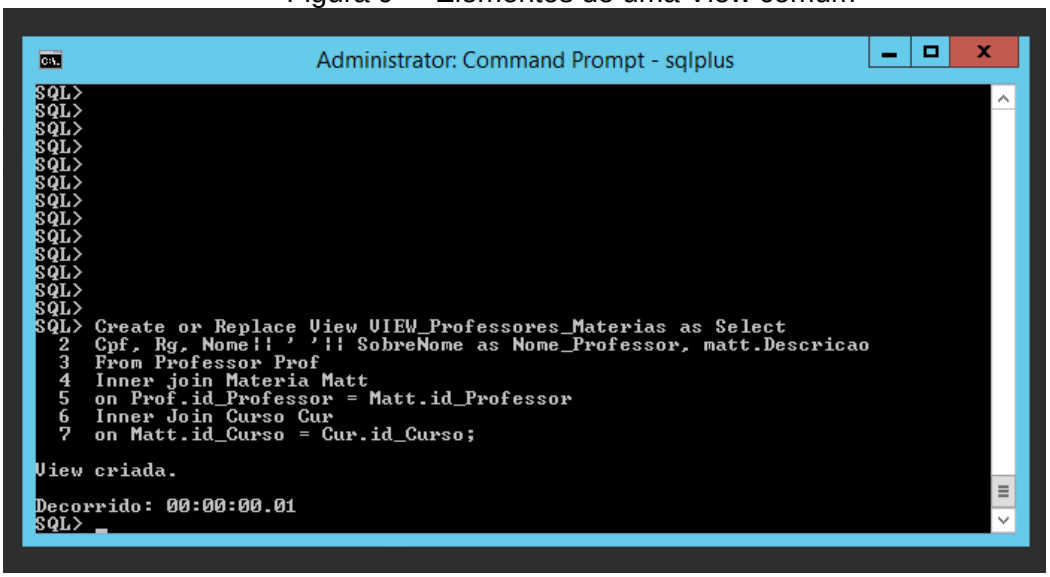
Figura 8 – Elementos de uma VM



Fonte: Ana Beatriz e Ricardo(2020)

Na Figura 9 Observamos o mesmo tipo de situação das imagens acima, porém com tabela contendo professor e matéria.

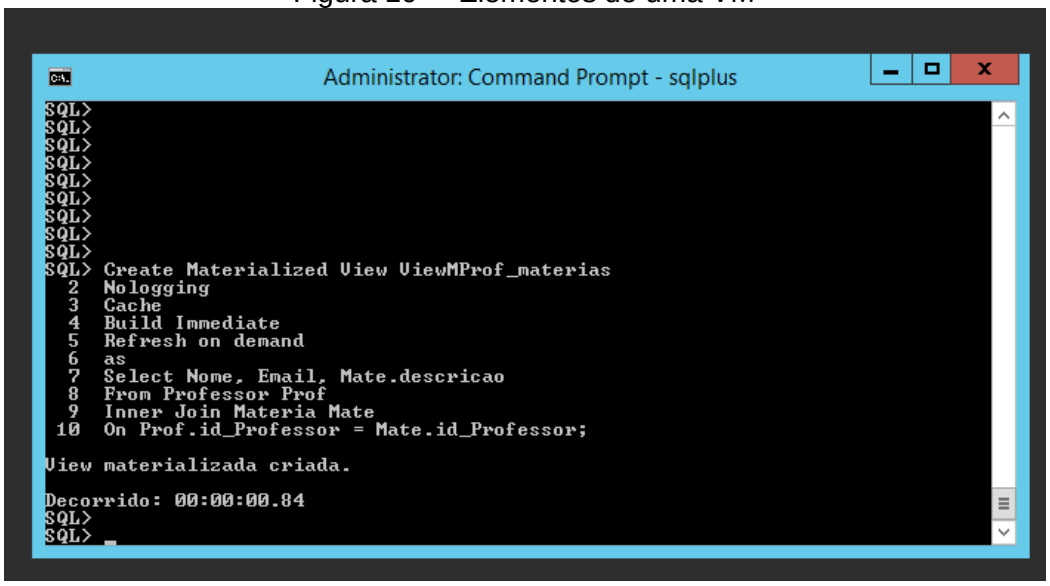
Figura 9 – Elementos de uma View comum



Fonte: Ana Beatriz e Ricardo(2020)

Figura 10, mesmas informações a serem executada. Podemos comparar que para a criação das duas tabelas a view comum está com apenas 01 centesimo de segundos, e a materializada com 84 centesimo de segundos.

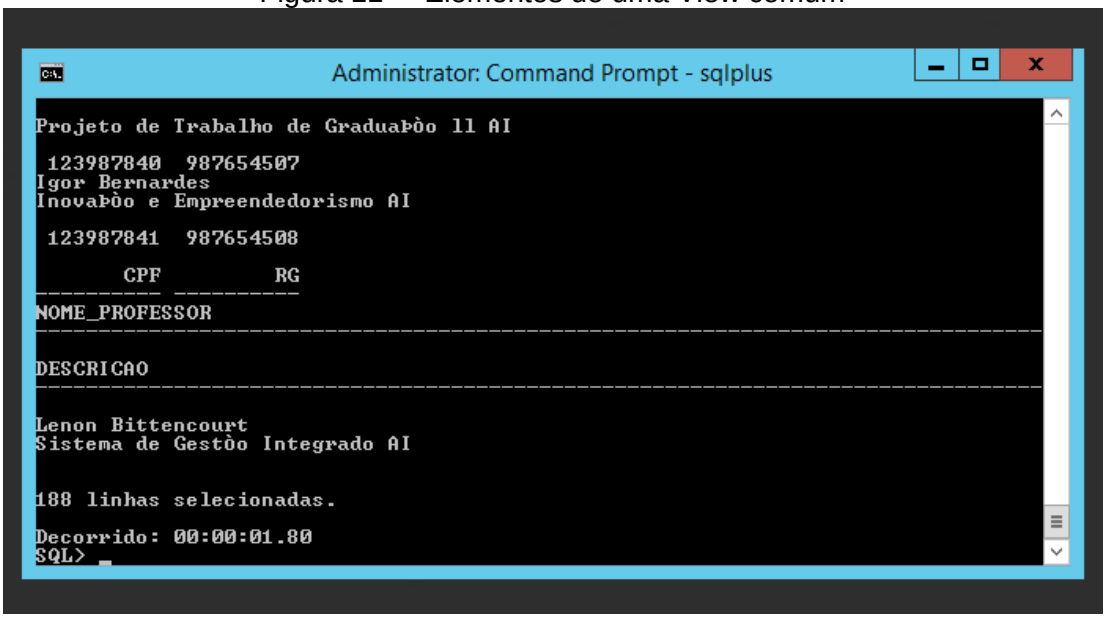
Figura 10 – Elementos de uma VM



Fonte: Ana Beatriz e Ricardo(2020)

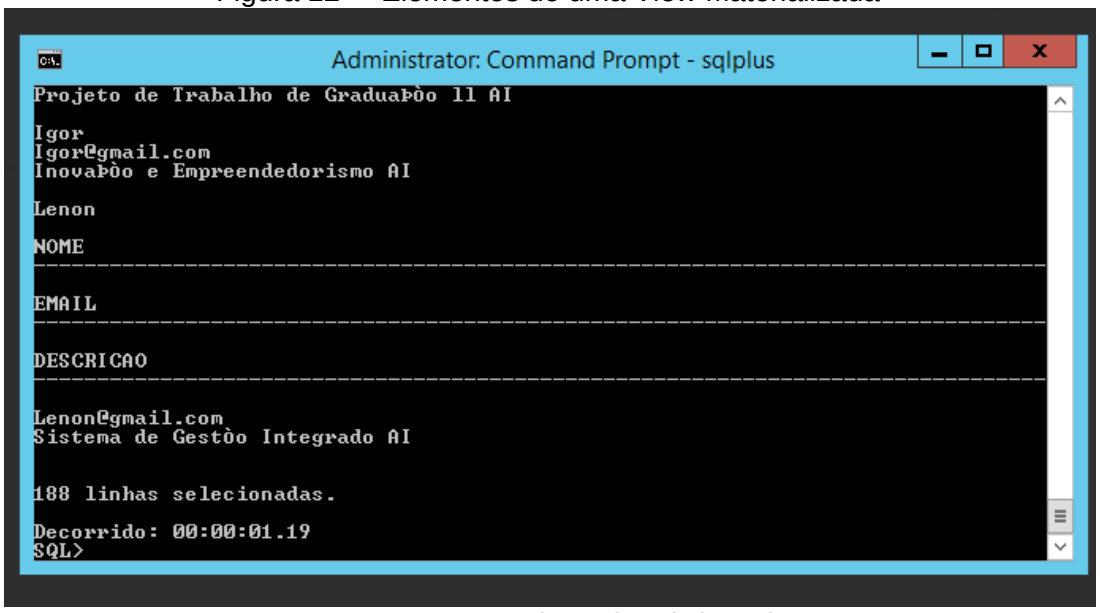
Na Figura 11 e 12 comparamos o tempo de execução da tabela professor e matéria,a viewcomum tem o tempo de 1.80 segundos e a materializada apenas com 1.19 segundos.

Figura 11 – Elementos de uma View comum



Fonte: Ana Beatriz e Ricardo(2020)

Figura 12 – Elementos de uma View Materializada

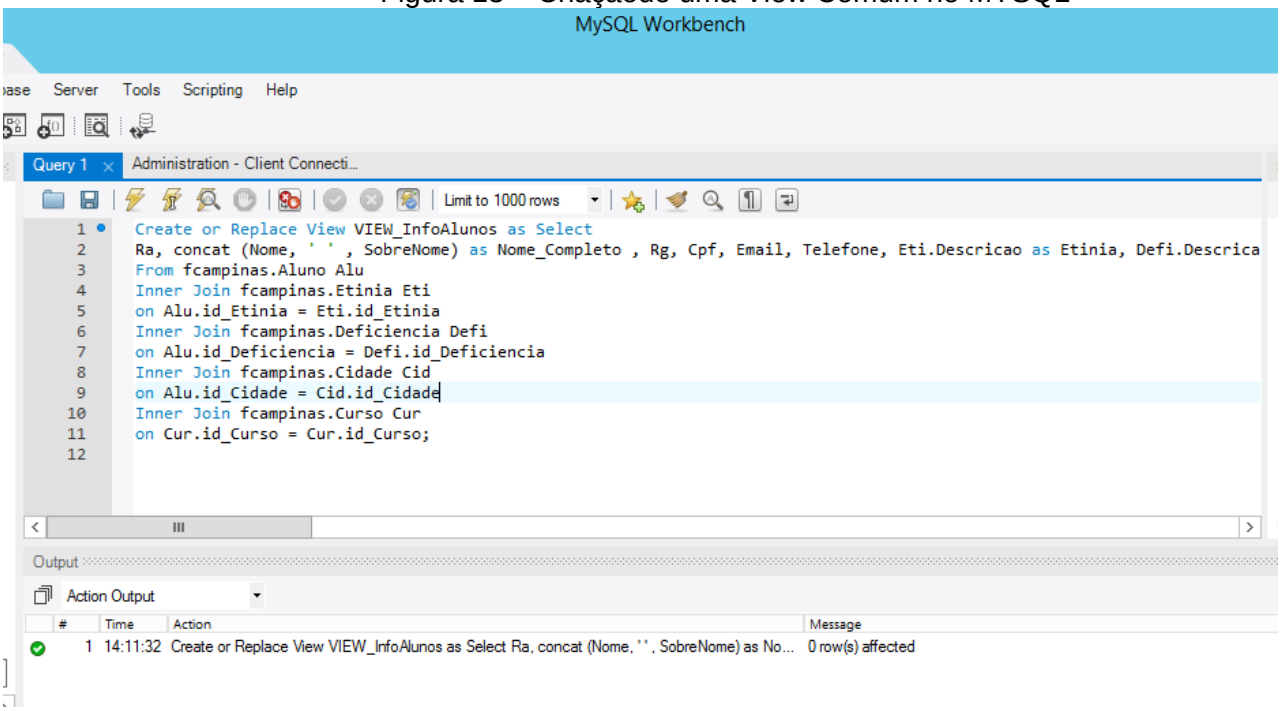


Fonte: Ana Beatriz e Ricardo(2020)

As figuras abaixo estão utilizando as mesmas informações das tabelas realizadas no banco Oracle nas figurasacima, porém essas estão no banco de dados MYSQL.Mostrando como são criadas e quaisforam seu tempo de criação e de execução dos selects.

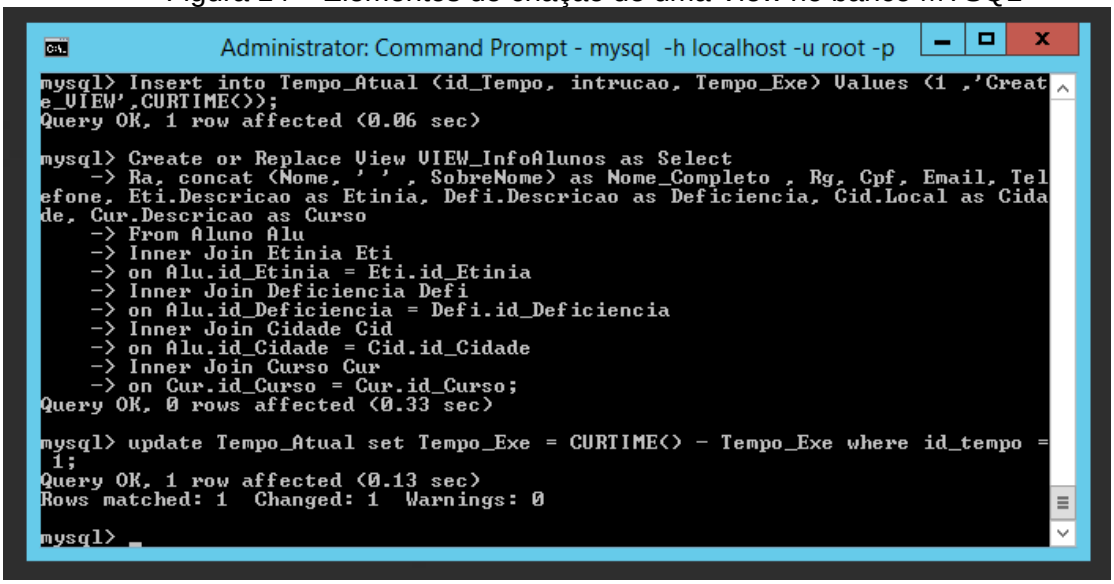
Figura 13 e 15 é a criação das views com a tabela info aluno, informações de cadastro do aluno na faculdade e criação da tabela com informação do professor e matéria.

Figura 13 – Criaçãode uma View Comum no MYSQL



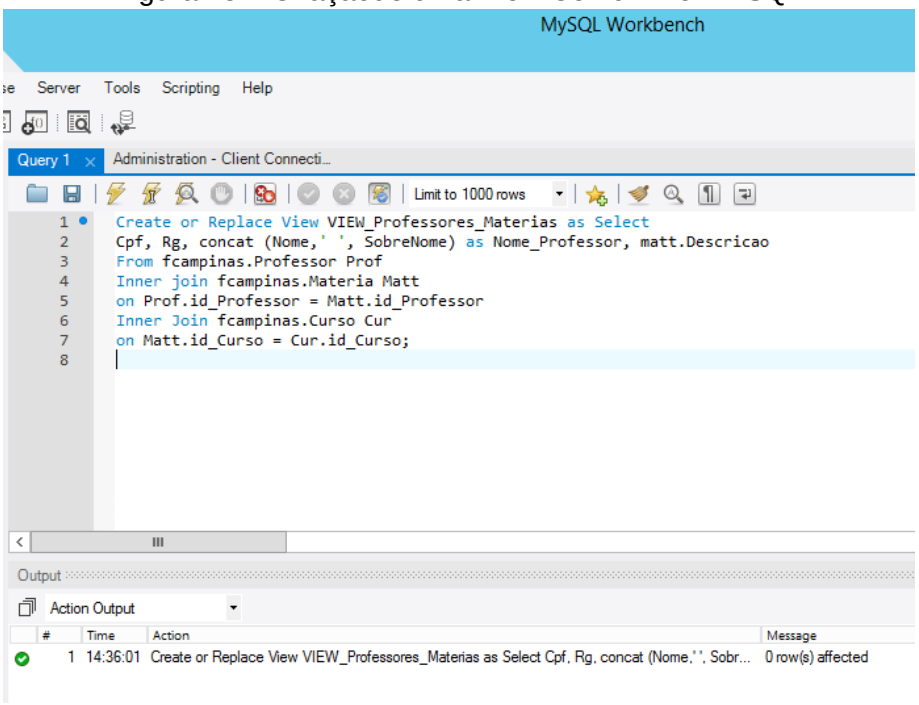
Fonte: Ana Beatriz e Ricardo(2020)

Figura 14 – Elementos de criação de uma View no banco MYSQL



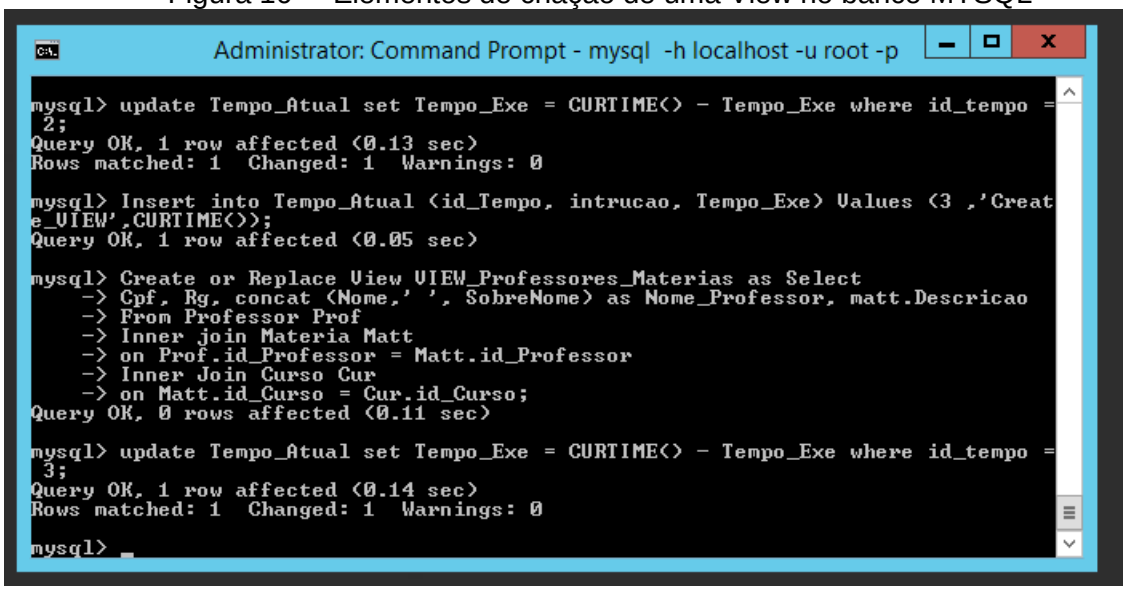
Fonte: Ana Beatriz e Ricardo(2020)

Figura 15 – Criaçãode uma View Comum no MYSQL



Fonte: Ana Beatriz e Ricardo(2020)

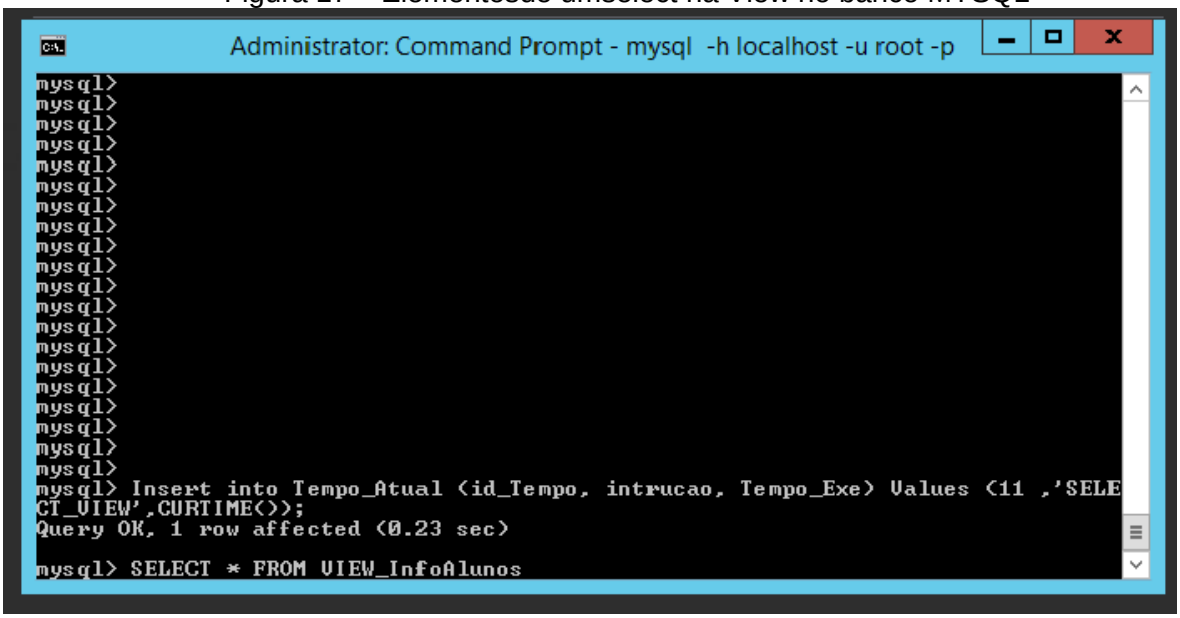
Figura 16 – Elementos de criação de uma View no banco MYSQL



Fonte: Ana Beatriz e Ricardo(2020)

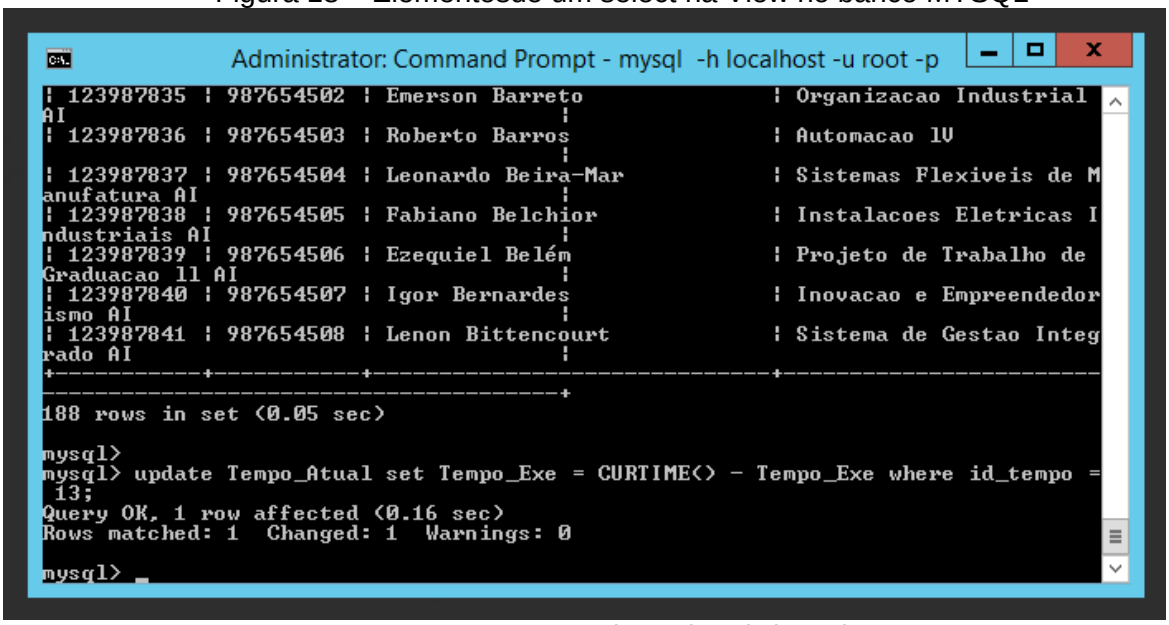
Nas Figuras 17 e 18 realizamos as execuções das tabelas criadas, transendo as informações contidas.

Figura 17 – Elementosde umselect na View no banco MYSQL



Fonte: Ana Beatriz e Ricardo(2020)

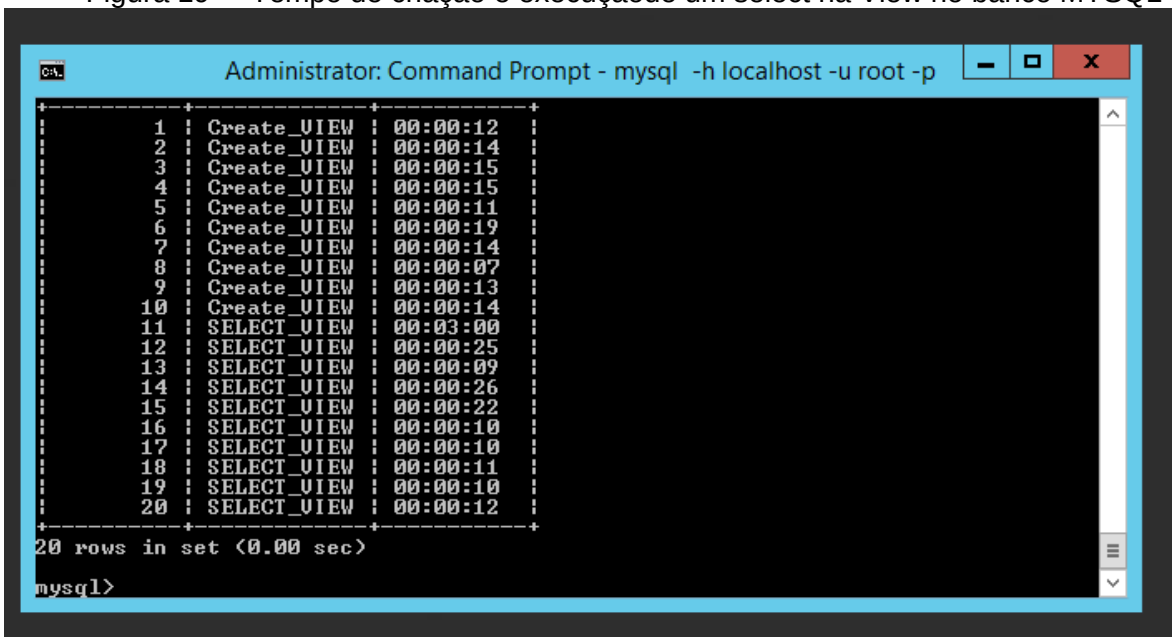
Figura 18 – Elementosde um select na View no banco MYSQL



Fonte: Ana Beatriz e Ricardo(2020)

Na Figura 19, mostramos o tempo realizado de criação e de execução das tabelas. O tempo de criação 1 (create_view) refere-se a viewinfo aluno da Figura 14 e o tempo de criação 3, refere-se a view professor e matéria da Figura 16. O tempo de execução do select 11 e 13 é da Figura 17 e 18.

Figura 19 – Tempo de criação e execuçãode um select na View no banco MYSQL



Fonte: Ana Beatriz e Ricardo(2020)

5. CONCLUSÃO

De acordo com os testes realizados, concluímos que a view comum cria as tabelas em um menor tempo, porém vimos que a materializada tem um melhor desempenho em executar os dados, ela trás os resultados mais rapidamente do que a comum. O que não leva a afirmarmos que uma é melhor do que a outra, pois considerando as duas, ambas são eficientes dependendo do seu objetivo. Cada uma delas tem suas qualidades em determinado momento e ambiente.

A view comum realiza a consulta no momento que o usuário fizer a consulta nela, com isso necessitando que o banco esteja sempre disponível, online.

A materializada é um pouco diferente, pois ela cria uma copia dos dados, portanto a VM continua funcionando mesmo se o banco de origem estiver indisponível. Mas ela não será uma boa opção para fins de emergência como uma agencia de banco que precisa sempre estar atualizando os seus dados. Lembrando também que a view materializada tem duas formas de ser atualizadas, refreshondemand, onde ela se atualiza quando for enviado um comando a ela e refreshoncommiton, atualiza cada vez que for dado um commit em uma transição.

6. REFERÊNCIAS

ALVES, G. *Qual a diferença entre View e MaterializedView.*2015. Disponível em:https://dicasdeprogramacao.com.br/qual-a-diferenca-entre-view-e-materialized-view/Acesso em:01out. 2018.

JANONES, R. *Entenda view3s em SQL, vantagens e desvantagens.* 2017 https://www.ramosdainformatica.com.br/banco_de_dados/vantagens-e-desvantagens-views-sql/ Acesso em: 04 jul. 2020.

MAIA, D. *Utilizando views, stored procedures e triggers.* 2014 https://pt.slideshare.net/maiamg/utilizando-views-stored-procedures-e-triggers Acesso: 04 jul. 2020.

PACIEVITCH, Y.*MySQL*. 2011. Disponível em:https://www.infoescola.com/informatica/mysql/Acesso em: 15out. 2018.

PULGA, S; FRANÇA, E; GOYA, M. **Banco de dados**implementação em SQL, PL/SQL e Oracle 11g. 2013. Cidade: São Paulo. Pearson Education do Brasil. 2014.

REIS, F. *O que é um Banco de Dados Relacional*. 2019. Disponível em: http://www.bosontreinamentos.com.br/bancos-de-dados/o-que-e-um-banco-de-dados-relacional/ Acesso em: 04 jul. 2020.

SILVA,D.*Banco de Dados*.2015. Disponível em:https://www.estudopratico.com.br/banco-de-dados/Acesso em: 15 abr. 2018.

SCUDERO,E.*Principais SGBDs do mercado global.*2016. Disponível em:https://becode.com.br/principais-sgbds/ Acesso em:10 set. 2018.