



**Universidade de São Paulo
Escola de engenharia de São Carlos
Instituto de Ciências Matemáticas e de Computação**

Games XP

Gerenciamento e Organização de Conteúdo do Evento

Subtema: Feira de Jogos

SSC: Base de Dados
Prof^a. Elaine Parros Machado de Sousa

Gabriel Caurin Corrêa - 9805081
Nilo Conrado Messias Alves Cangerana - 9805362

Sumário

Parte 1: Descrição do problema, requisitos de dados e projeto conceitual

Descrição geral.....	3
Consultas à Base de Dados.....	4
Modelo Entidade - Relacionamento.....	5

Parte 2: Mapeamento

Correções Parte 1.....	7
Modelo Relacional.....	7
Justificativas para o modelo relacional	8

Parte 3: Implementação SQL

Correções Parte 2.....	10
Implementação.....	10
Conclusão.....	18

1. Descrição do Problema

A Games XP é um parque de diversão temático de games, com atrações ao ar livre, jogos eletrônicos gratuitos, novas tecnologias e a maior tela de games do mundo.

O projeto a ser desenvolvido visa gerenciar unicamente os dados relacionados a **Feira de Jogos** da Games XP, dada a grandeza do evento. A proposta feita para a empresa responsável tem como objetivo reger o evento dando destaque aos **jogos** a serem apresentados e que estarão disponíveis em **estandes**, seus **estúdios** de desenvolvimento e **empresas** que os comandam, bem como **esquematizar as atrações disponíveis** nesta seção do evento com **desenvolvedores**, **personalidades** envolvidas com o desenvolvimento dos games e o **público** pagante.

Primeiramente identificaremos o coração do evento, os jogos. Os jogos são cadastrados no sistema com título, que é seu identificador, data de lançamento, plataformas em que ele está disponível, preço e se existe ou não DLC (conteúdo digital adicional que expande a experiência do jogo). Cada jogo possui um único estúdio em que foi desenvolvido, mas o estúdio pode se envolver em um ou mais projetos. O cadastro dos estúdios são feitos com nome, que é seu identificador único, data de fundação, número de funcionários e localização e um identificador se é indie ou não. Um estúdio é financiado por uma empresa, mas uma empresa pode financiar diversos estúdios. As empresas são identificadas pelo seu CNPJ, e possuem nome, data de fundação, localização da sede e o atual CEO da companhia.

O evento possuirá **estandes** onde os estúdios oferecerão testes de seus jogos por um período de tempo limitado a cada visitante da “barraca”. Os estandes serão enumerados para facilitar sua identificação em mapas do evento e terão uma área pré demarcada para montarem sua exposição. Toda empresa ou estúdio possui um estande, sendo que todo estande é pertencente a uma dessas entidades, mas nunca às duas ao mesmo tempo. Durante todo o evento os estandes estarão abertos ao **público** para visitação. Um estande pertencente a uma empresa é compartilhado entre todos os estúdios da mesma. Enquanto que estúdios independentes possuem seu próprio estande, garantido na aplicação e descrição do banco.

Apresentações serão realizadas durante o evento por empresas para divulgarem novidades relacionadas a sua marca. Elas são identificadas e diferenciadas pela data, hora e suas categorias, que são **duas**: 1) Publicidade envolvendo as empresas e seus produtos, onde o apresentador mostra o foco da empresa e seus jogos atuais, 2) Palestras realizadas pelas empresas, onde o palestrante desenvolve um tema atual na indústria dos games. Além das apresentações teremos também a realização de **entrevistas**, com data e hora previamente estabelecidas e diversos temas possíveis a serem discutidos por pessoas de dentro dos estúdios ou por personalidades que participam do desenvolvimento do jogo. As entrevistas, que são realizadas por desenvolvedores e personalidades, só podem ser realizadas uma vez por cada pessoa. As entrevistas são realizadas

em uma localização fixa no evento, e as apresentações são realizadas no palco principal do evento. Elas estarão abertas ao público que comparece ao evento.

Desenvolvedores dos estúdios, identificados por nome, CPF, contato(telefone e e-mail) e **formação**. Sua formação pode ser: 1)**programadores**, que possuem conhecimento sobre diversas linguagens de programação e possuem uma proficiência em resolver problemas de programação e, 2) **designers**, que possuem conhecimentos em softwares de modelagens gráficas e uma função no projeto. Desenvolvedores podem pertencer somente a um estúdio.

Personalidades são atores, dubladores ou escritores que realizam um trabalho no processo de criação do jogo, dependendo de sua função. São cadastrados no sistema com nome artístico e especialidade, que podem ser: 1)**contracenista**, que possui um papel no jogo atual que está trabalhando e também suas principais vertentes de atuação. 2)**Escritor**, que possui um gênero de escrita principal. 3)**Voice actor(dubladores de voz)**, que possui um papel no jogo atual e um timbre de voz específico.

Ao contrário dos desenvolvedores do estúdio, as personalidades diretamente ligadas ao jogo trabalham somente para o jogo em que possuem **contrato** de duração limitada, já que eles realizam uma função específica de um projeto. Os contratos são identificados pelo seu código e em seu cadastro é possível ver a duração do trabalho em questão e também sua descrição. O armazenamento de contratos anteriores permite que se tenha acesso a trabalhos passados da personalidade.

Também devemos endereçar aos jogos indies, seus devidos **patrocinadores** cadastrados com seu nome e identificados pelo seu CNPJ, que financiam projetos individualmente*. O financiamento dos jogos geram a sua divulgação, que é realizada pela mídia física e/ou digital(sites, revistas) e anúncios.

Por último, temos o **público** que comparecerá ao evento. Cada pessoa possui um crachá que dá acesso a diversos eventos da feira (apresentações, entrevistas e testes a jogos). O visitante em questão é identificado pelo número de seu crachá e, para finalizar, é também registrado seu CPF e nome.

*Entenda-se individualmente como um financiamento direto ao jogo, sem se relacionar com empresa e estúdio.

2. Consultas à Base de Dados

(1) Para inserção de uma nova apresentação no sistema, deve ser possível recuperar as datas e horários de todas as apresentações para evitar conflito, dado que não é possível duas apresentações no mesmo horário. Também é necessário checar horário e data de entrevistas, dado que o desenvolvedor entrevistado não pode realizar a entrevista no mesmo horário em que seu estúdio (supondo que o estúdio pertença a empresa) esteja realizando uma apresentação.

(2) Para a inserção de uma entrevista no sistema, a mesma checagem deve ser realizada. Duas entrevistas não podem ocorrer no mesmo horário e dia. Não pode haver conflito de horário de

apresentação de empresa com entrevista de um dado desenvolvedor de um estúdio pertencente a empresa em questão. Também deve ser possível recuperar dados de entrevistas já agendadas no sistema, pois um mesmo desenvolvedor/personalidade não pode realizar mais de uma entrevista.

(3) Dado um jogo, deve ser possível recuperar todos os eventos relacionados a ele na feira, tais como: estande no qual o jogo está disponível para teste e o tempo de teste, data e horário de entrevistas com os desenvolvedores/personalidades que trabalham no jogo, data e horário de apresentação caso o jogo pertença a uma empresa e patrocinador do jogo caso seja um jogo indie.

(4) Deve ser possível o agendamento de apresentações e entrevistas para um dado número de crachá do público. Não pode agendar apresentação e entrevista no mesmo horário.

(5) Dado o número de crachá do público, deve ser possível recuperar todos os eventos agendados para esta pessoa, tais como horário/data das apresentações e entrevistas que ela pode participar.

(6) Deve ser possível recuperar apresentações e entrevistas que serão realizadas em determinado dia ou com opção para pesquisar determinado horário e checar quais eventos estão disponíveis no dia/horário.

(7) Deve ser possível gerenciamento no banco de dados, com atualização ou remoção de apresentações e entrevistas agendadas. Para atualização, consideramos a mudança de horário, dia da mesma.

(8) Deve ser possível recuperar informações das personalidades que estarão presentes no evento, bem como o jogo ao qual estão participando e possíveis entrevistas que realizarão.

(9) Dado uma empresa, deve ser possível recuperar seu estande(numeração), data/hora de suas apresentações e jogos disponíveis na feira.

(10) Deve ser possível recuperar informação dos estúdios que estarão presentes no evento, dividindo em duas categorias: estúdios indies ou estúdios de empresas. Caso o estúdio seja indie, deve ser possível recuperar localização do estande, jogos disponíveis no evento e, caso possuir, patrocinador do jogo com suas devidas informações. Caso o estúdio seja de alguma empresa, deve ser possível recuperar localização do estande da empresa, jogos disponíveis no evento, e informações da empresa.

(11) Deve ser possível retornar a quantidade de pessoas que participaram de apresentações agrupadas por empresa. O resultado deve ser disposto em ordem decrescente para ver qual empresa possui mais público no evento.

(12) Deve ser possível recuperar, para cada patrocinador, a quantidade total de jogos que ele patrocina. E também deve ser recuperado o título do jogo mais patrocinado do evento.

3. Modelo Entidade - Relacionamento(MER)

4. Correções da Parte 1

O relacionamento Estande-Estúdio foi trocado de 1:1 para 1:N, já que um estande pode abrigar mais de um estúdio caso o mesmo pertença a uma empresa.

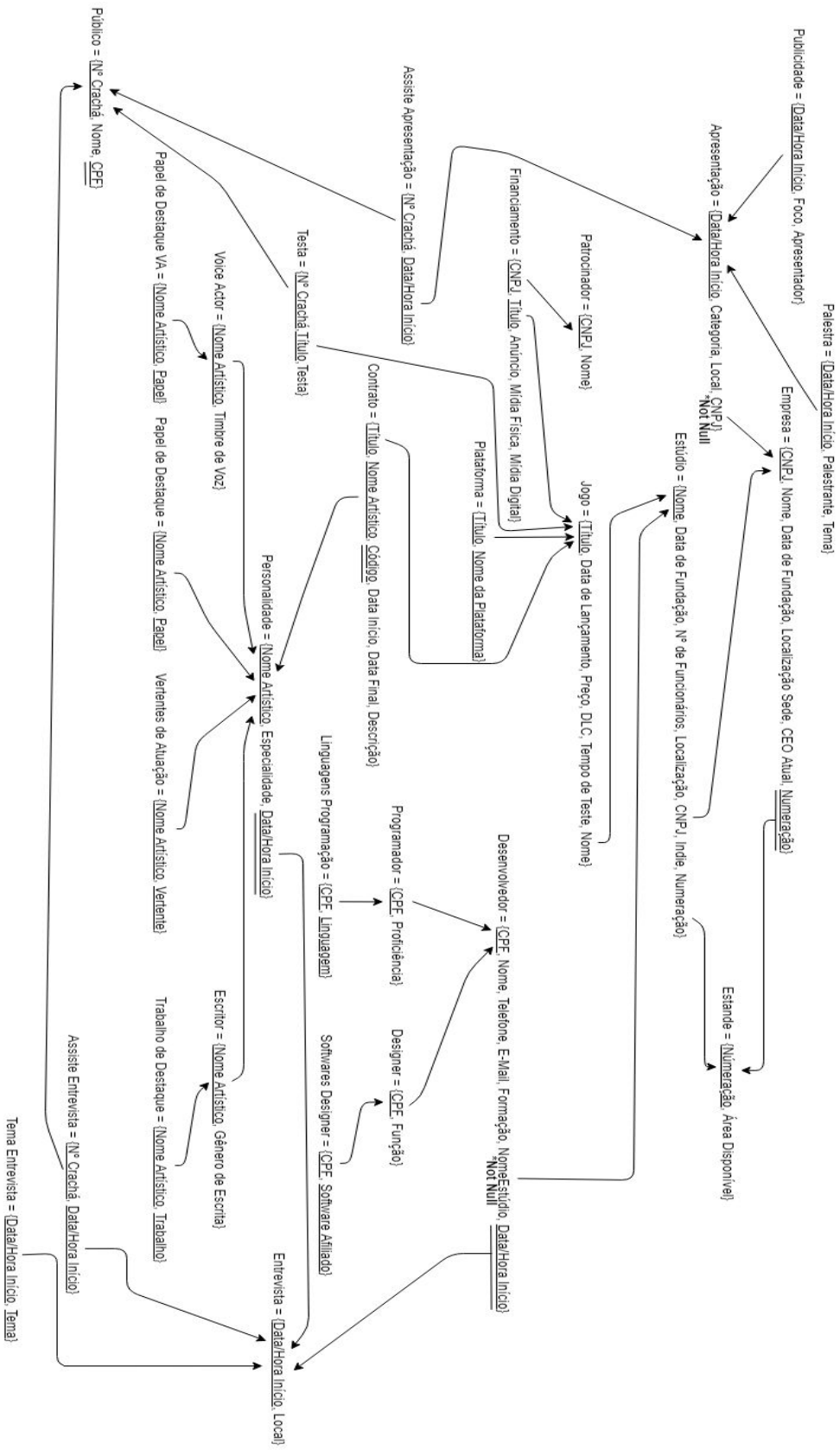
O relacionamento Público-Estande foi removido, já que não era necessário armazenar informações sobre qual pessoa passa por qual estande.

Nas generalizações de Apresentação e Desenvolvedor, foi trocado sua restrição de disjunção de sobreposição(O) para exclusão mútua(D).

Os relacionamentos de Público com Apresentação e Entrevistas são necessários para armazenar informações sobre a quantidade de pessoas que participam desses eventos, também é necessário manter um controle para saber quais eventos são os mais populares.

O relacionamento entre Jogo-Estande foi retirado e o tempo de jogo foi adicionado como atributo da entidade Jogo.

5. Modelo Relacional



6. Justificativas para o Modelo Relacional

Mapeamento de relacionamentos 1:1 :

No relacionamento 1:1 Empresa-Estande, a tabela de estande poderia ter uma coluna com o CNPJ da empresa e o CNPJ poderia ser uma chave secundária, garantindo o 1:1. No entanto, como estande possui um relacionamento com estúdio, e estúdio pode ou não ser de uma empresa, o campo CNPJ ficaria nulo para estúdios não relacionados com empresas e isso ocuparia espaço inutilizado. Logo, a tabela de empresa foi escolhida para conter a informação de estande no relacionamento 1:1.

Da mesma forma, a entidade entrevista possui relacionamento 1:1 com personalidades e desenvolvedores, logo, as tabelas de desenvolvedor e personalidades foram escolhidas para conter a informação de entrevistas como chave secundária. Caso entrevistas possuísse a coluna de desenvolvedores e personalidades, um elemento sempre estaria com valor nulo, devido ao fato de uma entrevista ser realizada por uma personalidade **ou** um desenvolvedor. Uma desvantagem dessa modelagem é que desenvolvedores e personalidades que não realizam entrevistas, terão o valor nulo no atributo de data/hora de início, ocupando espaço.

Mapeamento das generalizações:

A modelagem da generalização de Desenvolvedores escolhida não garante a participação total de desenvolvedores nas suas designadas formações específicas, porém foi escolhida essa montagem pois a entidade desenvolvedores participa de outros relacionamentos que são comuns para as entidades específicas. As tabelas específicas foram criadas para evitar que uma única tabela geral tivesse vários valores nulos, dado que a ocorrência da generalização é uma disjunção.

A mesma abordagem foi realizada para as generalizações de Personalidades e Apresentação. Da mesma forma, ela não garante a participação total e é necessário garantir em aplicação.

Para a modelagem de generalização garantir a participação total, era necessário criar uma tabela para cada entidade específica, e remover a tabela da generalização e, a partir disso, montar as relações de chaves estrangeiras para todas as relações que a entidade genérica participava, isso faria o diagrama do modelo relacional ficar muito mais confuso e repetitivo. Com uma tabela da entidade genérica, é possível modelar os relacionamentos com ela e deixar os específicos em suas respectivas tabelas, já que possuem atributos multivalorados que devem ser tratados com outra tabela.

Mapeamento de multivalorados:

Os atributos multivalorados de indicados no MER foram mapeados em uma nova tabela para cada atributo, de tal forma que seja possível adicionar qualquer informação necessária, sem qualquer limite. Caso fossem mapeados como uma nova coluna em sua respectiva tabela, seria necessário definir uma quantidade limitada destes valores. Se uma tupla não possui-se a quantidade necessária de informação para preencher todas as colunas novas, espaço seria desperdiçado e se uma tupla possui-se quantidade de atributos maior que a pré-programada, seria necessário cortar informação. Desta forma, a criação de uma nova tabela garante que toda a informação necessária seja armazenada sem ocupar mais espaço do que é preciso e sem perder informação.

Observações:

O atributo derivado Duração da entidade Entrevista não foi colocado no mapeamento pois ele é gerado em tempo de execução.

Na tabela Apresentação, a coluna de CNPJ de uma Empresa deve ser não nulo(NOT NULL) para garantir a participação total de Empresa no relacionamento Empresa-Apresentação.

Na tabela Desenvolvedor, a coluna de NomeEstúdio de um Estúdio deve ser não nulo(NOT NULL) para garantir a participação total de Desenvolvedor no relacionamento Desenvolvedor-Estúdio. NomeEstúdio é uma referência ao Nome do estúdio(que é único). O nome do atributo foi trocado para não haver confusão entre nome de desenvolvedor e nome do estúdio na mesma tabela Desenvolvedor.

7. Correções da Parte 2

O atributo de CPF de Público foi colocado como chave secundária.

A tabela de Contracenista foi deletada, pois estava com desperdício de espaço e a alternativa escolhida foi classificar o Contracenista através das duas tabelas de seus atributos. Neste caso, as tabelas Papel De Destaque e Vertentes da Atuação possuem os mesmos nomes artísticos, que são chave estrangeira de Personalidade. Assim, o Contracenista é identificado pelo atributo Especialidade de Personalidade e também pelos valores em cada tabela dos atributos.

Os atributos de data e hora de início e final das apresentações e das entrevistas foi trocado para um único atributo de data/hora de início, pois no SGBD da Oracle o tipo de dado DATE armazena dia e hora juntos.

O relacionamento entre público e jogo foi criado para armazenar os jogos que certa pessoa testa. O atributo do relacionamento Testa é booleano para saber se o jogo foi testado ou não pela pessoa.

Foi criado um atributo em estúdio para identificar se é indie ou não.

8. Implementação

A aplicação foi implementada em Java, feita no IDE Eclipse e utilizou-se um plugin chamado WindowBuilder para criação de uma interface gráfica funcional para realizar o acesso ao banco de dados. O projeto foi compilado e executado no Windows 10. A base de dados utilizada foi o Sql Developer da Oracle. Uma classe para conexão com o banco de dados foi criada (ProjetoBD - código fonte/src/GerarConexao.java) e foi implementado um método que realiza o acesso ao Sql Developer com o login e senha já colocados no código fonte. Para executar a aplicação é só clicar no AplicacaoBD.jar.

Implementação das tabelas:

Todas as tabelas geradas estão no arquivo GERAR TABELAS DA BASE.txt, com devida explicação dos códigos.

Alimentação da base dados:

Foi gerado um conjunto de testes para serem inseridos na base de dados. Os comandos para as inserções estão no arquivo ALIMENTACAO DA BASE.txt, com devida explicação dos códigos.

Implementação das buscas:

Foram escolhidas 6 operações de busca no banco de dados, que foram especificadas anteriormente, para serem implementadas. São elas:

(1) Para inserção de uma nova apresentação no sistema, deve ser possível recuperar as datas e horários de todas as apresentações para evitar conflito, dado que não é possível duas apresentações no mesmo horário. Também é necessário checar horário e data de entrevistas, dado que o desenvolvedor entrevistado não pode realizar a entrevista no mesmo horário em que seu estúdio (supondo que o estúdio pertença a empresa) esteja realizando uma apresentação.

Neste comando devem ser passados como parâmetros para a aplicação o CNPJ da empresa, Data/Hora de Início, Local, Palestrante ou Apresentador, Tema ou Foco e por fim selecionar se o tipo da apresentação é Publicidade ou Palestra. Fazendo isso, uma nova tupla de apresentação e palestra ou publicidade é inserida no banco de dados. Antes da inserção, é feita uma busca se algum desenvolvedor do estúdio da empresa faz entrevista no horário requerido. Se não, a inserção ocorre. Se sim, a inserção não ocorre e deve trocar a data. O script completo está na classe QueryUm.java linha 109(ProjetoBD - código fonte/src/QueryUm.java)

Primeiramente é usado um select para selecionar o CPF e Data/Início da entrevista do desenvolvedor, dado o CNPJ e a data inseridos na aplicação. Caso seja retornado algum valor não nulo, significa que existe uma entrevista no horário e o estúdio do desenvolvedor é da empresa logo, a inserção não pode ser feita. Se o valor retornado é nulo, o horário está livre e ocorre duas inserções (em Apresentação e Palestra ou Publicidade) com o comando insert.

```
PreparedStatement preparedStmtCheck = GerarConexao.c.prepareStatement("SELECT DISTINCT DEV.CPF,
TO_CHAR(DEV.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS'),EMP.CNPJ FROM
DESENVOLVEDOR DEV,ESTUDIO EST,EMPRESA EMP,ENTREVISTA ENT " +
"WHERE DEV.DATA_HORA_INICIO=? AND DEV.NOME_ESTUDIO=EST.NOME AND EST.CNPJ=?");
//checar data da apresentação com data de entrevista desenvolvedor da empresa
SimpleDateFormat fmt = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss"); //parâmetros vindos da interface
gráfica(textField e comboBox)
Date d = fmt.parse(textFieldData.getText()); //cria tipo data do java
```

```
preparedStmtCheck.setDate(1, new java.sql.Date(d.getTime())); //seta os parametros do
PreparedStatement
preparedStmtCheck.setString(2, textFieldCNPJ.getText());
ResultSet rsetCheck = preparedStmtCheck.executeQuery(); //Result Set recebido

if(rsetCheck.next()) //Checa se o result set está vazio. {
//nao pode inserir, result set tem conteúdo, ou seja há uma entrevista de desenvolvedor de empresa no horário
requerido.
JOptionPane.showMessageDialog(null, "Erro na inserção. Desenvolvedor da empresa faz entrevista no mesmo
horário requerido.");}
else{
//pode inserir, nao a entrevista de desenvolvedor de empresa no horário requerido.
PreparedStatement preparedStmt = GerarConexao.c.prepareStatement("INSERT INTO APRESENTACAO
(DATA_HORA_INICIO,CATEGORIA,LOCALIZACAO,CNPJ) VALUES (?, ?, ?, ?)"); //insere a apresentação.
```

```

:
PreparedStatement preparedStmt2 = GerarConexao.c.prepareStatement("INSERT INTO PALESTRA
(DATA_HORA_INICIO,TEMA,PALESTRANTE) VALUES (?,?,?)");//insere a palestra.
:
PreparedStatement preparedStmt2 = GerarConexao.c.prepareStatement("INSERT INTO PUBLICIDADE
(DATA_HORA_INICIO,FOCO,APRESENTADOR) VALUES (?,?,?)");//insere a publicidade.

```

(3) Dado um jogo, deve ser possível recuperar todos os eventos relacionados a ele na feira, tais como: estande no qual o jogo está disponível para teste e o tempo de teste, data e horário de entrevistas com os desenvolvedores/personalidades que trabalham no jogo, data e horário de apresentação caso o jogo pertença a uma empresa e patrocinador do jogo caso seja um jogo indie.

Neste comando deve ser passado como parâmetro o título do jogo e é retornado todas as apresentações e entrevistas relacionadas com o jogo dividida em duas tabelas, o estande de seu estúdio, seu tempo de teste e patrocinador do jogo caso seja indie. Para entrevistas, os temas são concatenados em uma única tupla por entrevista. O script completo está na classe QueryDois.java linha 123 (ProjetoBD - código fonte/src/QueryDois.java)

É utilizado um select para selecionar o estande do jogo, tempo de teste do jogo e se o jogo é indie não. Em seguida, dois selects são utilizados: um para selecionar informações da entrevista e outro para selecionar os temas das entrevistas dado sua data. É realizada uma comparação das duas tabelas pela Data/HoraInício para concatenar os temas na tupla de cada entrevista para melhor representação na tabela.

Após isso, é checado se o jogo é indie ou não e dependendo da sua categoria, é realizado um select diferente. Para os índies, é recuperado os patrocinadores do jogo. Para os não índies, é recuperado apresentações de sua empresa.

```

//comando SQL para buscar estande do jogo, tempo de teste e condição do jogo(indie ou não)
PreparedStatement preparedStmt = GerarConexao.c.prepareStatement("SELECT
J1.TEMPO_DE_TESTE,EST.INDIE,EST.NUMERACAO FROM JOGO J1,ESTUDIO EST WHERE
J1.NOME=EST.NOME AND J1.TITULO=?");
:
:
//Recuperar Entrevistas e Temas de jogos
//Comando SQL para buscar nome artístico ou nome de desenvolvedor/especialidade ou formação/dataInício da
entrevista/localização da entrevista
PreparedStatement preparedStmt2 = GerarConexao.c.prepareStatement("SELECT DISTINCT
NA.NOME_ARTISTICO , NA.ESPECIALIDADE , TO_CHAR(ENT.DATA_HORA_INICIO,'DD/MM/YYYY
HH24:MI:SS') AS DATA1, ENT.LOCALIZACAO "
+ "FROM PERSONALIDADE NA,ENTREVISTA ENT,JOGO J1,CONTRATO T1 "
+ "WHERE (ENT.DATA_HORA_INICIO=NA.DATA_HORA_INICIO) AND
(T1.NOME_ARTISTICO=NA.NOME_ARTISTICO) AND (T1.TITULO=J1.TITULO) AND (J1.TITULO=?) "
+ "UNION "

```

```

+ "SELECT DISTINCT DEV.NOME,
DEV.FORMACAO,TO_CHAR(ENT2.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS') AS DATA1,
ENT2.LOCALIZACAO FROM DESENVOLVEDOR DEV,ENTREVISTA ENT2,JOGO J2,ESTUDIO EST "
+ "WHERE ENT2.DATA_HORA_INICIO=DEV.DATA_HORA_INICIO AND
DEV.NOME_ESTUDIO=EST.NOME AND J2.NOME=EST.NOME AND J2.TITULO=? "
+ "ORDER BY DATA1");
:
:
//Comando SQL para buscar os temas das entrevistas
PreparedStatement preparedStmtTema = GerarConexao.c.prepareStatement("SELECT DISTINCT
TO_CHAR(ENT.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS') AS DATA1 , T_ENT.TEMA FROM
PERSONALIDADE NA,ENTREVISTA ENT,TEMA_ENTREVISTA T_ENT,JOGO J1,CONTRATO T1 " +
"WHERE (ENT.DATA_HORA_INICIO=NA.DATA_HORA_INICIO) AND
(T1.NOME_ARTISTICO=NA.NOME_ARTISTICO) AND (T1.TITULO=J1.TITULO) AND (J1.TITULO=?) AND
T_ENT.DATA_HORA_INICIO=ENT.DATA_HORA_INICIO " +
"UNION " +
"SELECT DISTINCT TO_CHAR(ENT2.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS') AS
DATA1,T_ENT2.TEMA FROM DESENVOLVEDOR DEV,ENTREVISTA ENT2,TEMA_ENTREVISTA
T_ENT2,JOGO J2,ESTUDIO EST " +
"WHERE ENT2.DATA_HORA_INICIO=DEV.DATA_HORA_INICIO AND
DEV.NOME_ESTUDIO=EST.NOME AND J2.NOME=EST.NOME AND J2.TITULO=? AND
T_ENT2.DATA_HORA_INICIO=ENT2.DATA_HORA_INICIO " +
"ORDER BY DATA1,TEMA",ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
:
:
//Recuperar Patrocinadores dos jogos indies
//Comando SQL para buscar CNPJ/Nome/Anuncio/Midia Digital/Midia Fisica de patrocinador
PreparedStatement preparedStmt3 = GerarConexao.c.prepareStatement("SELECT
PTR.CNPJ,PTR.NOME,FIN.ANUNCIO,FIN.MIDIA_DIGITAL,FIN.MIDIA_FISICA FROM PATROCINADOR
PTR,FINANCIAMENTO FIN,JOGO J1 " +
"WHERE PTR.CNPJ=FIN.CNPJ AND FIN.TITULO=J1.TITULO AND J1.TITULO=?");
:
:
//Recuperar Apresentacoes e Temas de Empresas
//Comando SQL para buscar informações das apresentacoes
PreparedStatement preparedStmt3 = GerarConexao.c.prepareStatement("SELECT
EMP.NOME,TO_CHAR(APR.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS') AS DATA1,
APR.CATEGORIA, PA_APR.PALESTRANTE,PA_APR.TEMA, APR.LOCALIZACAO FROM EMPRESA
EMP,APRESENTACAO APR,PALESTRA PA_APR,ESTUDIO EST,JOGO J1 " +
"WHERE APR.DATA_HORA_INICIO=PA_APR.DATA_HORA_INICIO AND APR.CNPJ=EMP.CNPJ AND
EST.CNPJ=EMP.CNPJ AND J1.NOME=EST.NOME AND J1.TITULO=? " +
"UNION " +
"SELECT EMP.NOME,TO_CHAR(APR.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS') AS DATA1,
APR.CATEGORIA, PU_APR.APRESENTADOR,PU_APR.FOCO, APR.LOCALIZACAO FROM EMPRESA
EMP,APRESENTACAO APR,PUBLICIDADE PU_APR,ESTUDIO EST,JOGO J1 " +
"WHERE APR.DATA_HORA_INICIO=PU_APR.DATA_HORA_INICIO AND APR.CNPJ=EMP.CNPJ AND
EST.CNPJ=EMP.CNPJ AND J1.NOME=EST.NOME AND J1.TITULO=? " +
"ORDER BY DATA1");

```

(9) Dado uma empresa, deve ser possível recuperar seu estande(numeração), data/hora de suas apresentações e jogos disponíveis na feira.

Neste comando deve ser passado como parâmetro o CNPJ da empresa e é retornado suas apresentações em uma tabela, jogos pertencentes a empresa em outra tabela e seu estande. Cada tupla de jogo contém a informação concatenada de suas plataformas disponíveis. O script completo está na classe Query3.java linha 74 (ProjetoBD - código fonte/src/Query3.java).

Um select é realizado inicialmente para recuperar estande da empresa. Em seguida, é buscado as apresentações da empresa com o select parametrizado pelo CNPJ. Por fim, dois selects são realizados para buscar jogos e suas plataformas disponíveis. Uma concatenação das plataformas é realizada por jogo, comparando o campo de título dos results sets retornados. Cada jogo é representado em uma tupla, com suas devidas plataformas.

```
//comando SQL para buscar estande da empresa
try {
PreparedStatement preparedStmt= GerarConexao.c.prepareStatement("SELECT EST.NUMERACAO FROM
EMPRESA EMP,ESTANDE EST WHERE EMP.NUMERACAO=EST.NUMERACAO AND EMP.CNPJ=?");
:
:

//Comando SQL para buscar informacoes de apresentacao
PreparedStatement preparedStmt2 = GerarConexao.c.prepareStatement("SELECT
TO_CHAR(APR.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS') AS DATA1, APR.CATEGORIA,
PA_APR.PALESTRANTE,PA_APR.TEMA, APR.LOCALIZACAO FROM EMPRESA EMP,APRESENTACAO
APR,PALESTRA PA_APR,ESTUDIO EST,JOGO J1 " +
"WHERE APR.DATA_HORA_INICIO=PA_APR.DATA_HORA_INICIO AND APR.CNPJ=EMP.CNPJ AND
EST.CNPJ=EMP.CNPJ AND EMP.CNPJ=? " +
"UNION " +
"SELECT TO_CHAR(APR.DATA_HORA_INICIO,'DD/MM/YYYY HH24:MI:SS') AS DATA1,
APR.CATEGORIA, PU_APR.APRESENTADOR,PU_APR.FOCO, APR.LOCALIZACAO FROM EMPRESA
EMP,APRESENTACAO APR,PUBLICIDADE PU_APR,ESTUDIO EST,JOGO J1 " +
"WHERE APR.DATA_HORA_INICIO=PU_APR.DATA_HORA_INICIO AND APR.CNPJ=EMP.CNPJ AND
EST.CNPJ=EMP.CNPJ AND EMP.CNPJ=? " +
"ORDER BY DATA1");
:
:

//Comando SQL para buscar jogos
PreparedStatement preparedStmt3 = GerarConexao.c.prepareStatement("SELECT
J1.TITULO,TO_CHAR(J1.DATA_DE_LANCAMENTO,'DD/MM/YYYY'),J1.PRECO,J1.DLC FROM JOGO
J1,ESTUDIO EST, EMPRESA EMP WHERE J1.NOME=EST.NOME AND EST.CNPJ = EMP.CNPJ AND
EMP.CNPJ=? ORDER BY J1.TITULO");
preparedStmt3.setString (1,textFieldCnpj.getText());
ResultSet rset3 = preparedStmt3.executeQuery();

//Comando SQL para buscar plataformas dos jogos
PreparedStatement preparedStmtP = GerarConexao.c.prepareStatement("SELECT
J1.TITULO,P1.NOME_DA_PLATAFORMA FROM JOGO J1,ESTUDIO EST, EMPRESA EMP,PLATAFORMA
P1 WHERE J1.NOME=EST.NOME AND EST.CNPJ = EMP.CNPJ AND EMP.CNPJ=? AND
```

```
P1.TITULO=J1.TITULO ORDER BY J1.TITULO",ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
preparedStmtP.setString (1,textFieldCnpj.getText());
ResultSet rsetP = preparedStmtP.executeQuery();
```

(10) Deve ser possível recuperar informação dos estúdios que estarão presentes no evento, dividindo em duas categorias: estúdios indies ou estúdios de empresas. Caso o estúdio seja indie, deve ser possível recuperar localização do estande, jogos disponíveis no evento e, caso possuir, patrocinador do jogo com suas devidas informações. Caso o estúdio seja de alguma empresa, deve ser possível recuperar localização do estande da empresa, jogos disponíveis no evento, e informações da empresa.

Neste comando é retornado todos estúdios do evento, junto com seus jogos e divididos em duas tabelas(indies ou não). Para os indies, o estúdio é repetido, mas o jogo é diferente, para ser possível representar os patrocinadores do jogo em questão na mesma tupla do jogo. Para os estúdios de empresa, cada estúdio é representado em uma tupla da tabela e seus jogos são concatenados na mesma linha. O código completo está na classe QueryQuatro.java linha 55 (ProjetoBD - código fonte/src/QueryQuatro.java).

Primeiramente é realizado um select para buscar informação dos estúdios de empresas e, posteriormente, outro select é realizado para buscar os jogos por estúdios. Os jogos são concatenados por estúdio para melhor representação na tabela. O mesmo é feito para estúdios indies e patrocinadores. No entanto, para evitar perda de quem patrocina qual jogo, o estúdio é repetido por jogo e seus patrocinadores são concatenados na tupla do jogo que patrocina.

```
//Comando para retornar informacao dos estudios de empresas
PreparedStatement preparedStmt= GerarConexao.c.prepareStatement("SELECT DISTINCT
EST.NOME,TO_CHAR(EST.DATA_FUNDACAO,'DD/MM/YYYY'),EST.NUM_FUNCIONARIOS,EST.LOCAL
IZACAO,EST.NUMERACAO,EMP.NOME FROM ESTUDIO EST,EMPRESA EMP WHERE
EST.CNPJ=EMP.CNPJ " +
"ORDER BY EST.NOME");
ResultSet rset = preparedStmt.executeQuery();
```

```
//Comando SQL para buscar jogos de estudios de empresas
PreparedStatement preparedStmtJ = GerarConexao.c.prepareStatement("SELECT EST.NOME,J1.TITULO FROM
ESTUDIO EST,JOGO J1 WHERE J1.NOME=EST.NOME AND EST.INDIE='0' " +
"ORDER BY EST.NOME",ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
ResultSet rsetJ = preparedStmtJ.executeQuery();
```

```
//Comando SQL para retornar informacao dos estudios indies
```

```

PreparedStatement preparedStmt2= GerarConexao.c.prepareStatement("SELECT
EST.NOME,TO_CHAR(EST.DATA_FUNDACAO,'DD/MM/YYYY'),EST.NUM_FUNCIONARIOS,EST.LOCAL
IZACAO,EST.NUMERACAO,J1.TITULO FROM ESTUDIO EST LEFT JOIN JOGO J1 ON
EST.NOME=J1.NOME WHERE EST.INDIE='1' " +
"ORDER BY J1.TITULO");
ResultSet rset2 = preparedStmt2.executeQuery();

```

//Comando SQL para retornar jogos e respectivos patrocinadores.

```

PreparedStatement preparedStmtP= GerarConexao.c.prepareStatement("SELECT J1.TITULO,PAT.NOME FROM
JOGO J1, PATROCINADOR PAT, FINANCIAMENTO F WHERE F.CNPJ=PAT.CNPJ AND
F.TITULO=J1.TITULO ORDER BY J1.TITULO",ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
ResultSet rsetP = preparedStmtP.executeQuery();

```

(11) Deve ser possível retornar a quantidade de pessoas que participaram de apresentações agrupadas por empresa. O resultado deve ser disposto em ordem decrescente para ver qual empresa possui mais público no evento.

Nesta pesquisa iremos buscar os nomes das empresas e contar a quantidade de pessoas que participaram de cada apresentação de cada empresa de forma que são retornados duas colunas na tabela, ordenados pela quantidade de pessoas decrescente. O código completo está na classe QueryCinco.java linha 42 (ProjetoBD - código fonte/src/QueryCinco.java).

Para essa busca inicialmente selecionamos a Tabela EMPRESA e realizamos uma junção interna (INNER JOIN) com a Tabela APRESENTACAO a partir das suas chaves primária e estrangeira, respectivamente. Em seguida, o mesmo processo foi repetido mais 2 vezes gerando quatro tabelas concatenadas (EMPRESA > APRESENTACAO > ASSISTE_APRESENTACAO > PUBLICO), onde finalmente podemos retornar o que desejamos: Nome da Empresa e o Total de pessoas que participaram de uma apresentação de tal empresa a partir de um contador.

//Comando SQL para realizar a contagem de público que assiste a apresentação por empresa.

```

try {
PreparedStatement preparedStmt = GerarConexao.c.prepareStatement("SELECT E.NOME ,
COUNT(P.NUM_CRACHA) AS CONTAGEM " +
"FROM EMPRESA E JOIN APRESENTACAO A " +
"      ON E.CNPJ = A.CNPJ " +
"      JOIN ASSISTE_APRESENTACAO A_A " +
"      ON A_A.DATA_HORA_INICIO = A.DATA_HORA_INICIO " +
"      JOIN PUBLICO P " +
"      ON p.num_cracha = a_a.num_cracha " +
"GROUP BY E.NOME " +
"ORDER BY CONTAGEM DESC,E.NOME");

```

```

ResultSet rset = preparedStmt.executeQuery();

```


(12)Deve ser possível recuperar, para cada patrocinador, a quantidade total de jogos que ele patrocina. E também deve ser recuperado o título do jogo mais patrocinado do evento.

Por fim, iremos fazer uma pesquisa em duas partes: Na primeira pesquisaremos todos os patrocinadores e retornar quantidade de jogos que cada um patrocina.

Nessa busca foi feito uma concatenação de 3 tabelas pelas restrições de chaves primária e estrangeira de cada uma a partir do método FROM JOIN ON. Após a junção das tabelas, separamos as tuplas em grupos de acordo com o nome do Patrocinador e retornamos o nome do patrocinador mais a quantidade de jogos que ele patrocina.

Já na segunda, faremos o inverso e retornaremos qual jogo possui mais patrocínios, limitando a busca em retornar apenas uma linha(com o maior valor encontrado).

Para essa outra busca, foi feito o mesmo processo, mas ao inverso na ordem de junção de tabelas. Separamos os grupos de acordo com o título do jogo e garantimos que em cada grupo a quantidade de patrocinadores deve ser maior que zero para não retornar nenhum valor vazio e irrelevante para essa busca. Foi feita uma consulta Aninhada para poder separar apenas a linha de valor maior de patrocinadores com a função ROWNUM.

O código completo está na classe QuerySeis.java linha 53 (ProjetoBD - codigo fonte/src/QuerySeis.java).

```
//Busca patrocinadores e conta quantidade de jogos patrocinados
PreparedStatement preparedStmt = GerarConexao.c.prepareStatement("SELECT P.NOME, COUNT(J.TITULO) " +
"FROM PATROCINADOR P JOIN FINANCIAMENTO F " +
"      ON P.CNPJ = F.CNPJ " +
"      JOIN JOGO J " +
"      ON J.TITULO = F.TITULO " +
"GROUP BY P.NOME " +
"ORDER BY P.NOME");
.
.
.
//Busca jogo mais patrocinado e quantidade de patrocínios
PreparedStatement preparedStmt2 = GerarConexao.c.prepareStatement("SELECT * " +
"FROM( " +
"SELECT J.TITULO, COUNT(P.NOME) " +
"FROM JOGO J JOIN FINANCIAMENTO F " +
"      ON J.TITULO = F.TITULO " +
"      JOIN PATROCINADOR P " +
"      ON P.CNPJ = F.CNPJ " +
"GROUP BY J.TITULO " +
"HAVING COUNT(P.NOME) > 0 " +
"ORDER BY COUNT(P.NOME) DESC) " +
"WHERE ROWNUM = 1");
```

9. Conclusão

Após um semestre realizando a confecção do trabalho prático, podemos perceber claramente ao finalizar as implementações SQL DDL e DML o quanto os modelos Entidade-Relacionamento e Relacional são de extrema importância para uma Base de Dados estável e consistente. A divisão do trabalho em três partes, além de evitar o acúmulo de trabalho em um só, ajudou os integrantes do grupo a se manterem à par do desenrolamento da matéria.

A implementação feita em Java foi relativamente simples uma vez que já somos familiarizados com os conceitos da linguagem Java Orientado à Objeto e nos deu uma aplicação “*interdisciplinar*” entre Banco de Dados e Programação Orientada à Objetos, fato que não é recorrente a maioria das matérias do curso.