

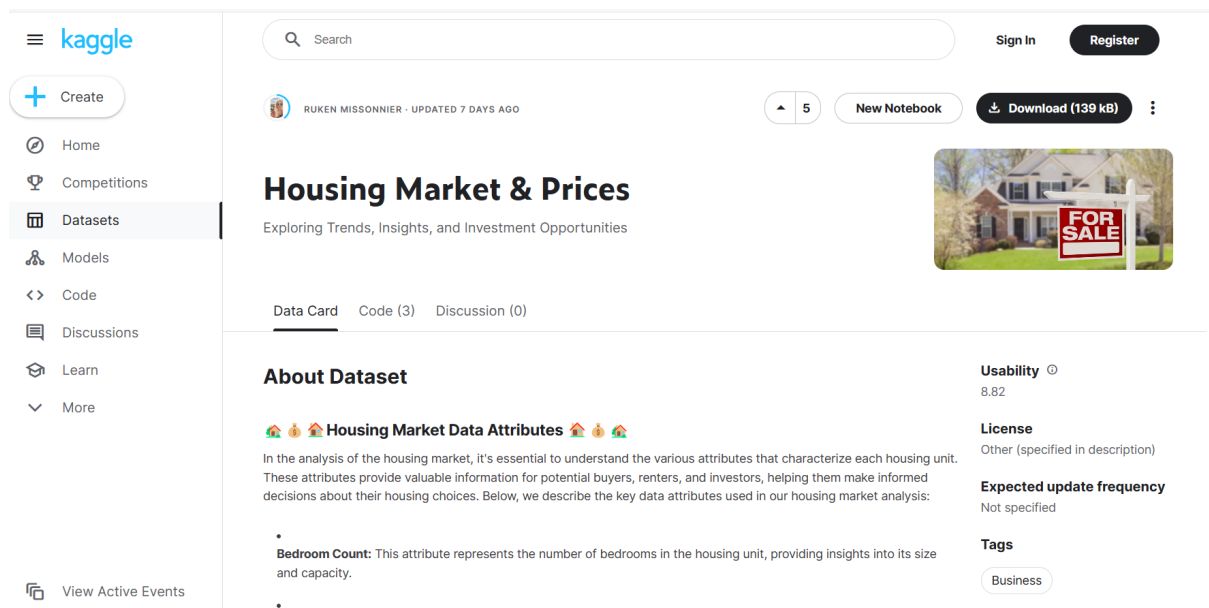
MVP - Sprint III

Engenharia de Dados

Aluna: Nathalia Azevedo

Busca pelos dados

Os dados escolhidos para desenvolvimento desta análise foi um banco de dados do mercado imobiliário retirado do Kaggle (Fonte: [Housing Market & Prices | Kaggle](#)).

The image is a screenshot of the Kaggle website showing the 'Housing Market & Prices' dataset. On the left is a navigation sidebar with links like Home, Competitions, Datasets, Models, Code, Discussions, Learn, and More. The main content area has a search bar at the top and a header for the dataset titled 'Housing Market & Prices' with the subtitle 'Exploring Trends, Insights, and Investment Opportunities'. Below the header are tabs for 'Data Card', 'Code (3)', and 'Discussion (0)'. The 'Data Card' is selected, showing an 'About Dataset' section with a description of the data attributes and a 'Usability' score of 8.82. There is also a 'License' section and a 'Tags' section with 'Business' as a tag. A small image of a house with a 'FOR SALE' sign is visible on the right side of the dataset header.

Metadados

Atributos de dados do mercado imobiliário

Na análise do mercado imobiliário, é essencial entender os diversos atributos que caracterizam cada unidade habitacional. Esses atributos fornecem informações valiosas para potenciais compradores, locatários e investidores, ajudando-os a tomar decisões informadas sobre suas escolhas de moradia. Abaixo, descrevemos os principais atributos de dados usados em nossa análise do mercado imobiliário:

- **bedroom_count (contagem de quartos):** Esse atributo representa o número de dormitórios da unidade habitacional, fornecendo informações sobre seu tamanho e capacidade.

- **m2_area (metros quadrados líquidos m² líquidos):** Os metros quadrados líquidos referem-se ao espaço interior útil total dentro da unidade habitacional, excluindo áreas comuns como corredores e escadas. Ele quantifica o tamanho do imóvel.
- **center_distance (distância do centro):** Esse atributo mede a distância da unidade habitacional da área central ou central de uma cidade. É uma métrica valiosa para potenciais compradores ou locatários avaliarem a proximidade com as comodidades e atividades urbanas.
- **metro_distance (distância do metrô):** A distância do metrô indica a distância entre a unidade habitacional e a estação de metrô ou metrô mais próxima. Essas informações são particularmente úteis para indivíduos que dependem do transporte público para seu deslocamento diário.
- **floor (andar):** O atributo piso especifica o nível ou o andar da unidade habitacional dentro do edifício, oferecendo informações sobre sua colocação e acessibilidade dentro da estrutura.
- **age (idade):** A idade do imóvel representa o número de anos desde a sua construção ou reforma. Desempenha um papel crucial na avaliação do estado da propriedade e potenciais requisitos de manutenção.
- **price (preço):** Preço é o custo associado à compra ou aluguel da unidade habitacional. É um fator fundamental para os indivíduos tomarem decisões de moradia e pode ser influenciado por vários atributos, como número de quartos, tamanho, localização e idade.

Objetivo

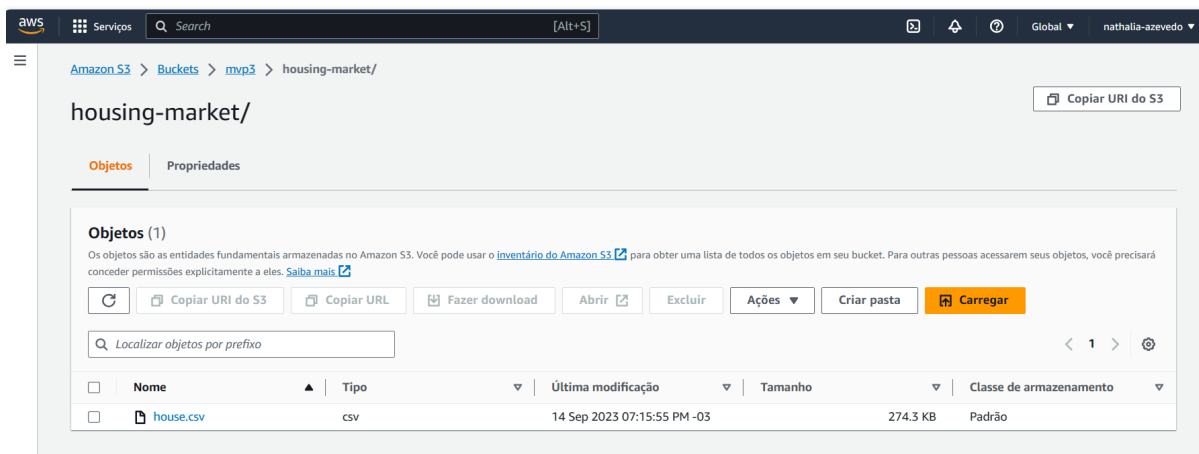
A partir de dados sobre o mercado imobiliário, com o objetivo de verificar qual atributo mais encarece o preço da casa, analisaremos as seguintes questões:

- 1 - Qual o preço médio das casas?
- 2 - Entre as casas no centro, qual é a média de preço das casas?
- 3 - Quantas casas de 2 quartos temos?
- 4 - Onde fica e quantos quartos tem a casa mais cara?
- 5 - E a mais barata?

Coleta

Os dados foram baixados para máquina local e inseridos manualmente em um *bucket* criado com nome de *mvp3*, do S3 da AWS.

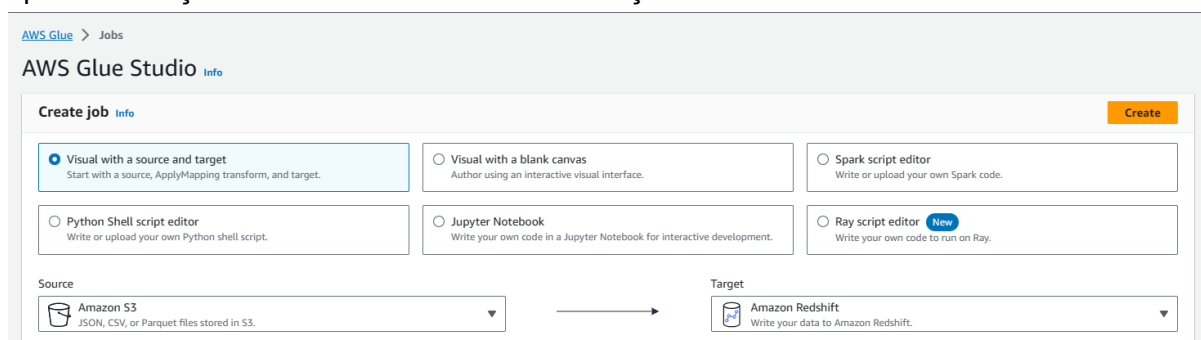
A inserção do arquivo foi feita dentro de uma pasta com nome de *housing-market* dentro do *Bucket* *mvp3*.



Modelagem e Carga

O ETL foi realizado utilizando o serviço AWS Glue. Através da sua interface visual jobs foram criadas e conectadas as seguintes etapas.

Após a criação do Bucket fazemos a criação do Job com a ferramenta utilizada.



E então configuramos a relação do Bucket com o Redshift, adicionando uma transformação entre o bucket e o redshift e definindo os nodes de referência destas etapas de transformação e data target.

Na etapa 1, o S3 bucket “Data Source - S3 Bucket”, foram realizadas as configurações para extrair os dados da fonte, no caso a pasta “housing-market” do *bucket* “mvp3”. Nesta etapa, configuramos o tipo do arquivo fonte como CSV, o separador como vírgula.

housing-market-jobs

Job has not been saved Try new UI Actions Save Run

Visual 1 Script Job details 1 Runs Data quality New Schedules Version Control

Data source - S3 bucket S3 bucket

Transform - Change Schema Change Schema

Data target - Amazon Redshift Amazon Redshift

Data source properties - S3

Output schema Data preview

Name S3 bucket

S3 source type Info

S3 location Choose a file or folder in an S3 bucket.

Data Catalog table

S3 URL s3://mvp3/housing-market/ View Browse S3

Recursive Read files in all subdirectories.

Data format CSV

Delimiter Comma (,)

Escape character - optional Enter a character to use for escaping

Na etapa 2, “Transform - Change Schema”, realizamos a etapa de transformação dos dados, convertendo o tipo de dados dos atributos “bedroom_count”, “floor” e “age” para *int*, e do “net_sqm”, “center_distance”, “metro_distance” e “price” para *float*. Também renomeei o atributo “net_sqm” para “m2_area”.

Transform

Output schema Data preview

Node parents Choose which nodes will provide inputs for this one.

Choose one or more parent node

S3 bucket S3 - DataSource

Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
bedroom_count	bedroom_count	int	<input type="checkbox"/>
net_sqm	m2_area	float	<input type="checkbox"/>
center_distance	center_distance	float	<input type="checkbox"/>
metro_distance	metro_distance	float	<input type="checkbox"/>
floor	floor	int	<input type="checkbox"/>
age	age	int	<input type="checkbox"/>
price	price	float	<input type="checkbox"/>

Na etapa 3, “Data Target - Amazon Redshift”, configuramos os conectores para o redshift e testamos a conexão com nome de mvp-glue-redshift, escolhemos o *schema* como public.

housing-market-jobs

⚠ Job has not been saved

🔄 Try new UI

Actions

Save

Run

Visual

Script

Job details

Runs

Data quality New

Schedules

Version Control

+

🗑

Data source - S3 bucket

S3 bucket

✔

🔄

Transform - Change Sche...

Change Schema

✔

📄

Data target - Amazon Re...

Amazon Redshift

⚠

🔍

🗑

🔄

📄

Data target properties - Amazon Redshift

Output schema

Data preview

Name

Amazon Redshift

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent node

Change Schema

ApplyMapping - Transform

Redshift access type

☒ Direct data connection - recommended

☐ Glue Data Catalog tables

Redshift connection

Choose the AWS Glue connection for Amazon Redshift, or [create a new connection](#)

Choose your Amazon Redshift connection

Connection

Database

▶ Performance and security

AWS Glue

>

Connectors

>

Create connection

Create connection

Info

Connection properties

Info

Name

Enter a unique name for your connection.

mvp-glue-redshift

Connection type

Amazon Redshift

☐ Require SSL connection

The connection will fail if it's unable to connect over SSL.

Description - optional

Descriptions can be up to 2048 characters long.

Connection access

Database instances

Provisioned Amazon Relational Database Service instances.

default-workgroup

CloudShellComentáriosIdioma

✔ Configuração bem-sucedida do Amazon Redshift Serverless

Revise suas definições de configuração. Para consultar dados, acesse o editor de consultas.

🔔 Try new Amazon Redshift features in preview

Create a workgroup with preview features. Production use of the workgroup is not supported. Use this workgroup for testing only.

Amazon Redshift Serverless

Painel do Serverless

Informações

Visão geral do namespace

Informações

Dados do namespace da sua conta

Total de snapshots

0

Compartilhamentos de dados na minha conta

0

Compartilhamentos de dados que exigem autorização

0

Compartilhamentos de dados

0

Namespaces / Grupos de trabalho

Informações

Namespace	Status	Grupo de trabalho	Status
default-namespace	✔ Available	default-workgroup	✔ Available

Amazon Redshift Serverless > Configuração do namespace > default-namespace

default-namespaceInformações

Ações ▼Alterar senha de administradorConsultar dados

Informações gerais

Namespace

default-namespace

Status

Available

Nome do usuário administrador

admin

Namespace ID

81478a47-f640-4158-a07e-99ccef7ee783

Data de criação

September 15, 2023, 16:54 (UTC-03:00)

Nome do banco de dados

dev

Namespace ARN

arn:aws:redshift-serverless:us-east-1:482664703545:namespace/81478a47-f640-4158-a07e-99ccef7ee783

Storage used

74 MB

Total table count

-

WorkgroupBackup de dadosSegurança e criptografiaUnidades de compartilhamento de dadosTags

Workgroup name

Set up compute resources for your workgroup.

Ações ▼

Workgroup

default-workgroup

Status

Available

Identity and Access Management (IAM)

Pesquisar no IAM

Painel

Gerenciamento de acesso

Grupos de usuários

Usuários

Funções

Políticas

Provedores de identidade

Configurações da conta

Relatórios de acesso

Analizador de acesso

Regras de arquivamento

Analísadores

Configurações

Relatório de credenciais

Atividade da organização

Políticas de controle de serviço

Allows Glue to call AWS services on your behalf.

Resumo

Editar

Data de criação

September 15, 2023, 17:17 (UTC-03:00)

ARN

arn:aws:iam::482664703545:role/MvpGlue

Última atividade

-

Duração máxima da sessão

1 hora

PermissõesRelações de confiançaEtiquetasConsultor de acessoRevogar sessões

Políticas de permissões (1)Informações

Você pode anexar até 10 políticas gerenciadas.

Pesquisar

Filtrar por Tipo

Todos os tipos

< 1 >

Nome da política

Tipo

Associar entidades

AdministratorAccess

Gerenciadas pela AWS - função de trabalho

1

Limite de permissões (não definido)

Após configurar a conexão, voltamos ao redshift no visual job, e então precisamos definir a tabela que ainda não foi criada.

Data target properties - Amazon Redshift 1 | Output schema | Data preview

Choose which nodes will provide inputs for this one.

Choose one or more parent node

Change Schema X
ApplyMapping - Transform

Redshift access type

☒ Direct data connection - *recommended*

☐ Glue Data Catalog tables

Redshift connection

Choose the AWS Glue connection for Amazon Redshift, or [create a new connection](#).

mvp-glue-redshift ↻

Connection

View properties

Database
dev

Schema

Choose your Amazon Redshift schema.

public ↻

Table

Search and enter the name of the source Amazon Redshift table.

↻

▼ Performance and security

S3 staging directory

Choose an S3 location for temporarily staging data in the format s3://bucket/prefix/object/ with a trailing slash (/).

Para criarmos a tabela, vamos ao Amazon Redshift e vamos no editor para consultar os dados, abrindo a seguinte janela com uma tela em que podemos criar a tabela com a linguagem SQL. Após utilizar o comando create table para criar a tabela housing_market inicialmente.

Redshift query editor v2

Untitled 1 x

Run Limit 100 Explain Isolated session Serverless: default-workgroup dev Schedule

```
1 create table public.housing_market (bedroom_count int, m2_area numeric, center_distance numeric, metro_distance numeric, floor int, age int, price numeric)
```

Result 1

Export Chart

Summary

Returned rows: 0

Elapsed time: 313ms

Result set query:

```
create table public.housing_market (bedroom_count int, m2_area numeric, center_distance numeric, metro_distance numeric, floor int, age int, price numeric)
-- RequestID=a5280332-0b9b-4238-9179-3f812822a942; TraceID=1-6504cabe-5efad7723b4258f17087d21a
```


house-market

Last modified on 15/09/2023, 18:28:40

Try new UI

Actions

Save

Run

Visual

Script

Job details

Runs

Data quality New

Schedules

Version Control

Data source - S3 bucket

S3 bucket

Transform - Change Sche...

Change Schema

Data target - Amazon Re...

Amazon Redshift

Data target properties - Amazon Redshift

Output schema

Data preview

Redshift connection

Choose the AWS Glue connection for Amazon Redshift, or [create a new connection](#).

mvp-glue-redshift

Connection

View properties

Database

dev

Schema

Choose your Amazon Redshift schema.

public

Table

Search and enter the name of the source Amazon Redshift table.

housing_market

Handling of data and target table

APPEND (insert) to target table

AWS Glue will append data to existing columns of the table and discard any extra columns.

MERGE data into target table

AWS Glue will either update or append data to the table based on a set of conditions.

TRUNCATE target table

Same as Append, except AWS Glue will first clear the contents of the table.

DROP and recreate target table

AWS Glue will delete and recreate the table with the schema from the source data.

house-market

Last modified on 15/09/2023, 18:28:40

Try new UI

Actions

Save

Run

Visual

Script

Job details

Runs

Data quality New

Schedules

Version Control

Job runs (1/1) Info

Last updated (UTC) September 15, 2023 at 21:31:30

View details

Stop job run

Table View

Card View

Filter job runs by property

Run status

Retries

Start time

End time

Duration

Capacity (DPUs)

Worker type

Glue version

Succeeded

0

09/15/2023 18:28:47

09/15/2023 18:30:22

1 m 17 s

2 DPUs

G.1X

4.0

09/15/2023 18:28:47

CloudWatch continuous logs

Driver logs

Driver and executor log streams

23/09/15 21:30:11 INFO LogPusher: stopping

23/09/15 21:30:11 INFO ProcessLauncher: postprocessing

23/09/15 21:30:10 INFO DefaultJDBCWrapper\$: End JDBC call 5

23/09/15 21:30:00 INFO DefaultJDBCWrapper\$: Begin JDBC call 5

23/09/15 21:30:00 INFO DefaultJDBCWrapper\$: End JDBC call 4

23/09/15 21:30:00 INFO DefaultJDBCWrapper\$: Begin JDBC call 4

23/09/15 21:30:00 INFO DefaultJDBCWrapper\$: End JDBC call 3

23/09/15 21:30:00 INFO RedshiftWriter: Creating table within Redshift: "public"."housing_market"

23/09/15 21:30:00 INFO DefaultJDBCWrapper\$: Begin JDBC call 3

23/09/15 21:30:00 INFO DefaultJDBCWrapper\$: End JDBC call 2

Após todos os dados inseridos podemos rodar o job.

house-market

Last modified on 15/09/2023, 18:28:40

Try new UI

Actions

Save

Run

Visual

Script

Job details

Runs

Data quality New

Schedules

Version Control

Job runs (1/1) Info

Last updated (UTC) September 15, 2023 at 21:30:00

View details

Stop job run

Table View

Card View

Filter job runs by property

Run status

Retries

Start time

End time

Duration

Capacity (DPUs)

Worker type

Glue version

Running

0

09/15/2023 18:28:47

-

55 s

2 DPUs

G.1X

4.0

09/15/2023 18:28:47

Stop job run

Job name

Id

Run status

Glue version

house-market

jr_beeeb8b0eb441dd6dbd19100232390b5df41ddf4de0fe17d06bd92fa75b73725a

Running

4.0

Retry attempt number

Start time

End time

Start-up time

Initial run

15 de setembro de 2023 18:28:47

-

0

Execution time

Last modified on

Trigger name

Security configuration

55 seconds

15 de setembro de 2023 18:28:51

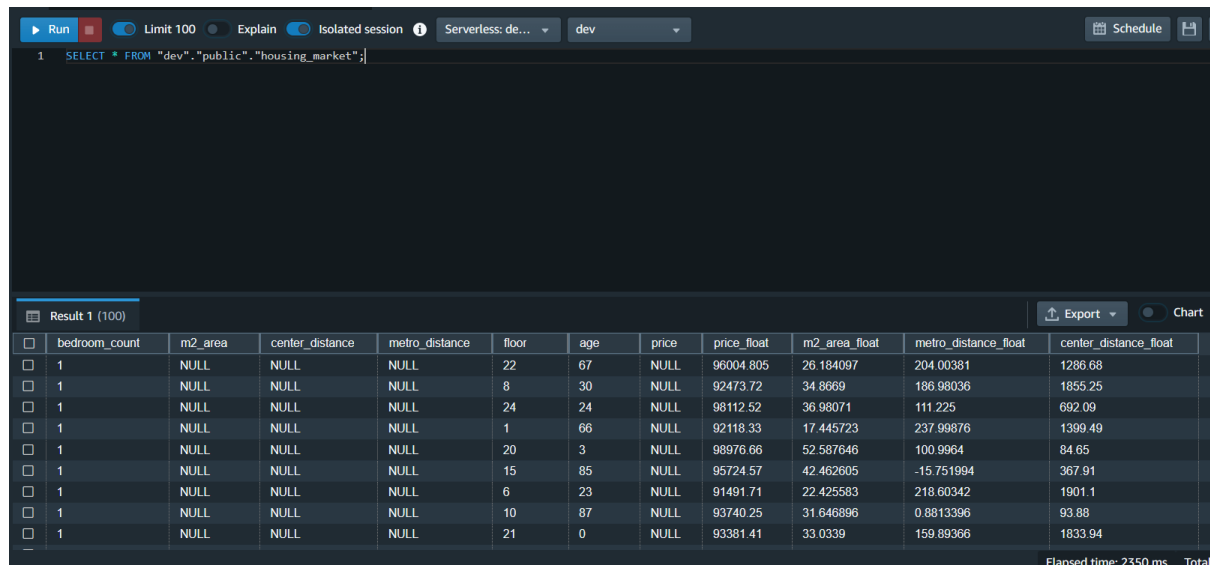
-

-

CloudShell

© 2023 Amazon Web Services, Inc. ou suas afiliadas. Privacidade Termos Preferências de cookies

Após rodar o job, no qual insere os dados do arquivo csv, no banco de dados no redshift, podemos verificar novamente na query com a linguagem SQL como se apresentam os dados.

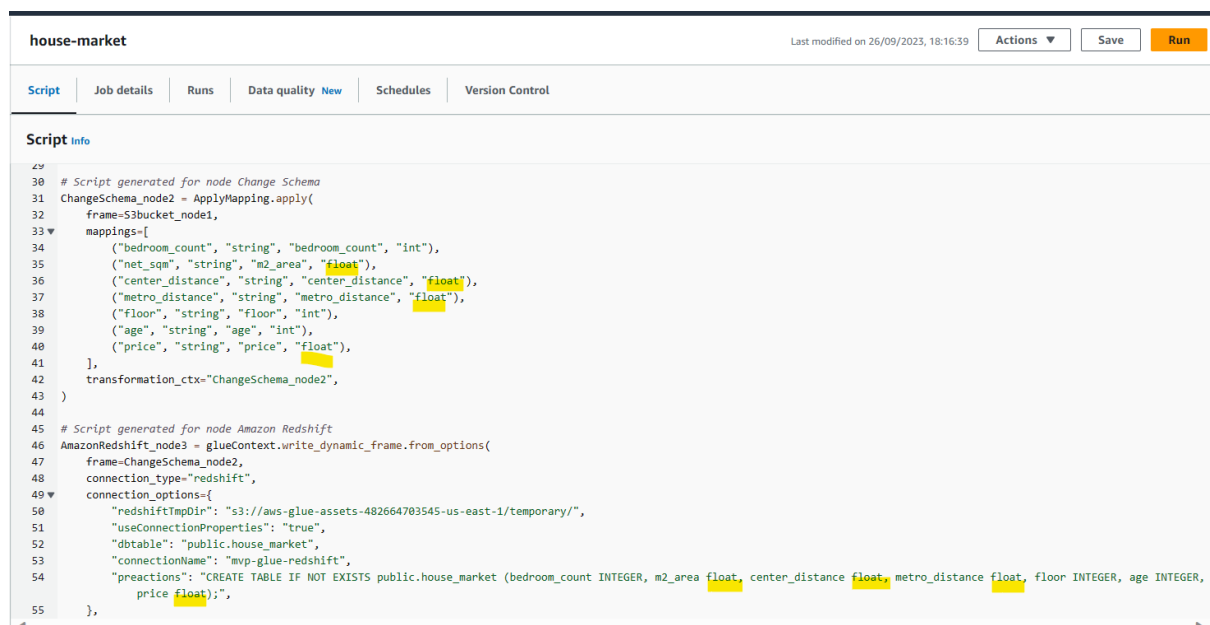


The screenshot shows a SQL query interface with a query editor at the top and a results table below. The query is `SELECT * FROM "dev"."public"."housing_market";`. The results table, titled "Result 1 (100)", displays 10 rows of data. The original columns are `bedroom_count`, `m2_area`, `center_distance`, `metro_distance`, `floor`, `age`, and `price`. The `m2_area`, `center_distance`, and `metro_distance` columns contain `NULL` values. New columns, `m2_area_float`, `metro_distance_float`, and `center_distance_float`, contain the decimal values for these fields. The `price` column also contains decimal values. The interface includes buttons for "Run", "Limit 100", "Explain", "Isolated session", "Serverless: de...", "dev", "Schedule", and "Export". The elapsed time is 2350 ms.

	bedroom_count	m2_area	center_distance	metro_distance	floor	age	price	price_float	m2_area_float	metro_distance_float	center_distance_float
1	1	NULL	NULL	NULL	22	67	NULL	96004.805	26.184097	204.00381	1286.68
1	1	NULL	NULL	NULL	8	30	NULL	92473.72	34.8669	186.98036	1855.25
1	1	NULL	NULL	NULL	24	24	NULL	98112.52	36.98071	111.225	692.09
1	1	NULL	NULL	NULL	1	66	NULL	92118.33	17.445723	237.99876	1399.49
1	1	NULL	NULL	NULL	20	3	NULL	98976.66	52.587646	100.9964	84.65
1	1	NULL	NULL	NULL	15	85	NULL	95724.57	42.462605	-15.751994	367.91
1	1	NULL	NULL	NULL	6	23	NULL	91491.71	22.425583	218.60342	1901.1
1	1	NULL	NULL	NULL	10	87	NULL	93740.25	31.646896	0.8813396	93.88
1	1	NULL	NULL	NULL	21	0	NULL	93381.41	33.0339	159.89366	1833.94

Podemos perceber que após rodar o job da forma que foi configurado pelo Change Schema no visual, gerou uma tabela na qual a coluna original não englobou os valores *float*, sendo preenchida por *NULL*. E foram geradas novas colunas com os valores decimais.

Para contornar este problema, eu recorri ao script gerado no glue, e editei o código para gerar as tabelas e carregar os valores, percebi que a configuração que estava sendo editada para as colunas `m2_area`, `center_distance` e `metro_distance` era `REAL` ao invés de *float*.



The screenshot shows a Glue job script for a job named "house-market". The script is divided into two sections: "Script Info" and "Script generated for node Change Schema". The "Script Info" section shows the job details, including the last modified date (26/09/2023, 18:16:39) and buttons for "Actions", "Save", and "Run". The "Script generated for node Change Schema" section shows the code for the job, including the mapping of columns and the transformation context. The script is as follows:

```
29
30 # Script generated for node Change Schema
31 ChangeSchema_node2 = ApplyMapping.apply(
32     frame=S3bucket_node1,
33     mappings=[
34         ("bedroom_count", "string", "bedroom_count", "int"),
35         ("net_sqm", "string", "m2_area", "float"),
36         ("center_distance", "string", "center_distance", "float"),
37         ("metro_distance", "string", "metro_distance", "float"),
38         ("floor", "string", "floor", "int"),
39         ("age", "string", "age", "int"),
40         ("price", "string", "price", "float"),
41     ],
42     transformation_ctx="ChangeSchema_node2",
43 )
44
45 # Script generated for node Amazon Redshift
46 AmazonRedshift_node3 = glueContext.write_dynamic_frame.from_options(
47     frame=ChangeSchema_node2,
48     connection_type="redshift",
49     connection_options={
50         "redshiftTmpDir": "s3://aws-glue-assets-482664703545-us-east-1/temporary/",
51         "useConnectionProperties": "true",
52         "dbtable": "public.house_market",
53         "connectionName": "mvp-glue-redshift",
54         "preactions": "CREATE TABLE IF NOT EXISTS public.house_market (bedroom_count INTEGER, m2_area float, center_distance float, metro_distance float, floor INTEGER, age INTEGER, price float);",
55     },
```

E então pude verificar que consegui carregar os dados da forma correta.

Run Limit 100 Explain Isolated session Serverless: de... dev Schedule

```
1 SELECT * FROM "dev"."public"."house_market";
```

Result 1 (100)

	bedroom_count	m2_area	center_distance	metro_distance	floor	age	price
1	1	26.184097	1286.68	204.00381	22	67	96004.805
1	1	34.8669	1855.25	186.98036	8	30	92473.72
1	1	36.98071	692.09	111.225	24	24	98112.52
1	1	17.445723	1399.49	237.99876	1	66	92118.33
1	1	52.587646	84.65	100.9964	20	3	98976.66
1	1	42.462605	367.91	-15.751994	15	85	95724.57
1	1	22.425583	1901.1	218.60342	6	23	91491.71
1	1	31.646896	93.88	0.8813396	10	87	93740.25
1	1	33.0339	1833.94	159.89366	21	0	93381.41
1	1	27.88348	1384.89	145.17003	14	12	93503.28
1	1	1.7831576	1296.94	123.92678	16	77	89214.57

Análise

Agora voltando ao objetivo inicial iremos responder as perguntas propostas.

- 1) Qual o preço médio das casas?

```
SELECT avg(price) FROM "dev"."public"."house_market";
```

Result 1 (1)

	avg
	95733.95645859085

- 2) Entre as casas no centro, qual é a média de preço das casas?

```
SELECT avg(price) FROM public.house_market WHERE center_distance < 250
```

Result 1 (1)

	avg
	98488.32136147295

- 3) Quantas casas de 2 quartos temos?

```
SELECT      count(bedroom_count)      as      casas_2qts      FROM
public.house_market WHERE bedroom_count = 2
```

Result 1 (1)	
<input type="checkbox"/>	casas_2qts
<input type="checkbox"/>	1019

4) Quais as informações tem a casa mais cara?

```
SELECT * FROM public.house_market WHERE price IN (SELECT
max(price) from public.house_market)
```

Result 1 (1)							
<input type="checkbox"/>	bedroom_count	m2_area	center_distance	metro_distance	floor	age	price
<input type="checkbox"/>	17	750.9716	402.62	40.98593	1	80	118134.77

5) E a mais barata?

```
SELECT * FROM public.house_market WHERE price IN (SELECT
min(price) from public.house_market)
```

Result 1 (1)							
<input type="checkbox"/>	bedroom_count	m2_area	center_distance	metro_distance	floor	age	price
<input type="checkbox"/>	1	9.469599	1959.05	96.21162	1	83	84153.484

Percebi que os atributos de andar e idade praticamente não influenciam no preço da casa, que o maior influenciador é quantos metros quadrados a casa tem, o que tem relação direta com a quantidade de quarto.