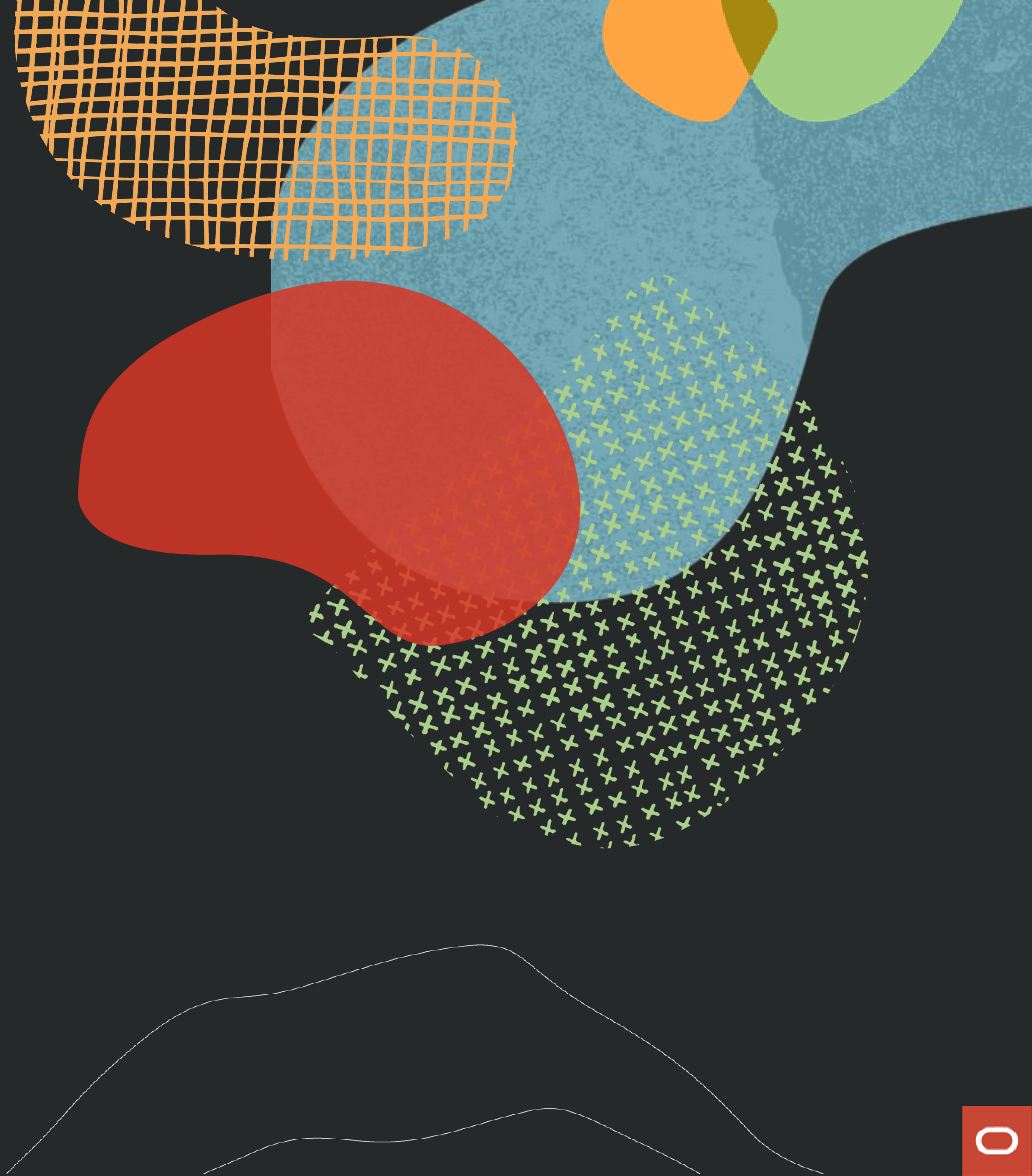
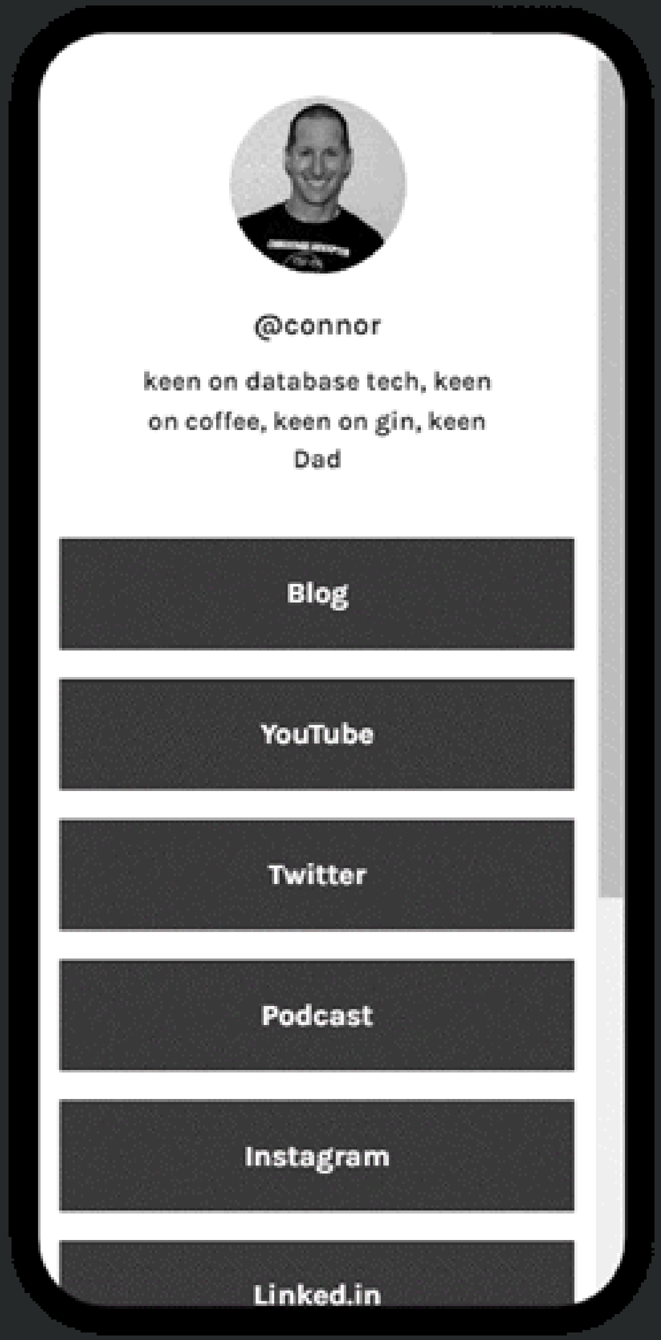
The Future of Data and App Dev



Connor McDonald

Database Advocate



@connor

keen on database tech, keen on coffee, keen on gin, keen Dad

Blog

YouTube

Twitter

Podcast

Instagram

Linked.in

@connor_mc_d

https://linktr.ee/connor

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle’s products may change and remains at the sole discretion of Oracle Corporation.

what is the future of data and app dev?

"Build more systems!"

... with more complexity

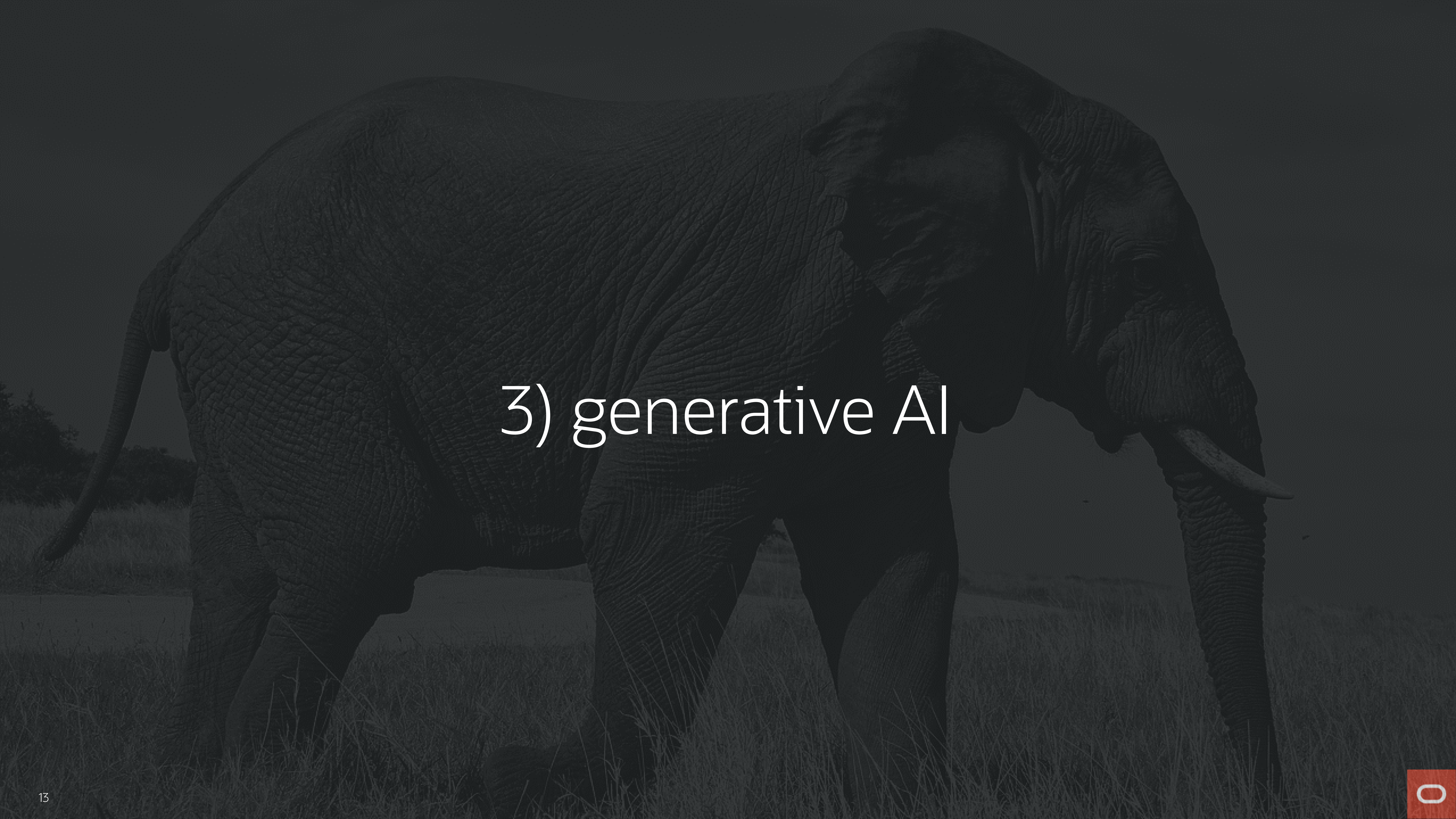
... and do it faster"

how?

3 core objectives

1) generate data for usage

2) generate apps not code them



3) generative AI

four interested parties

Data Professional

Developer

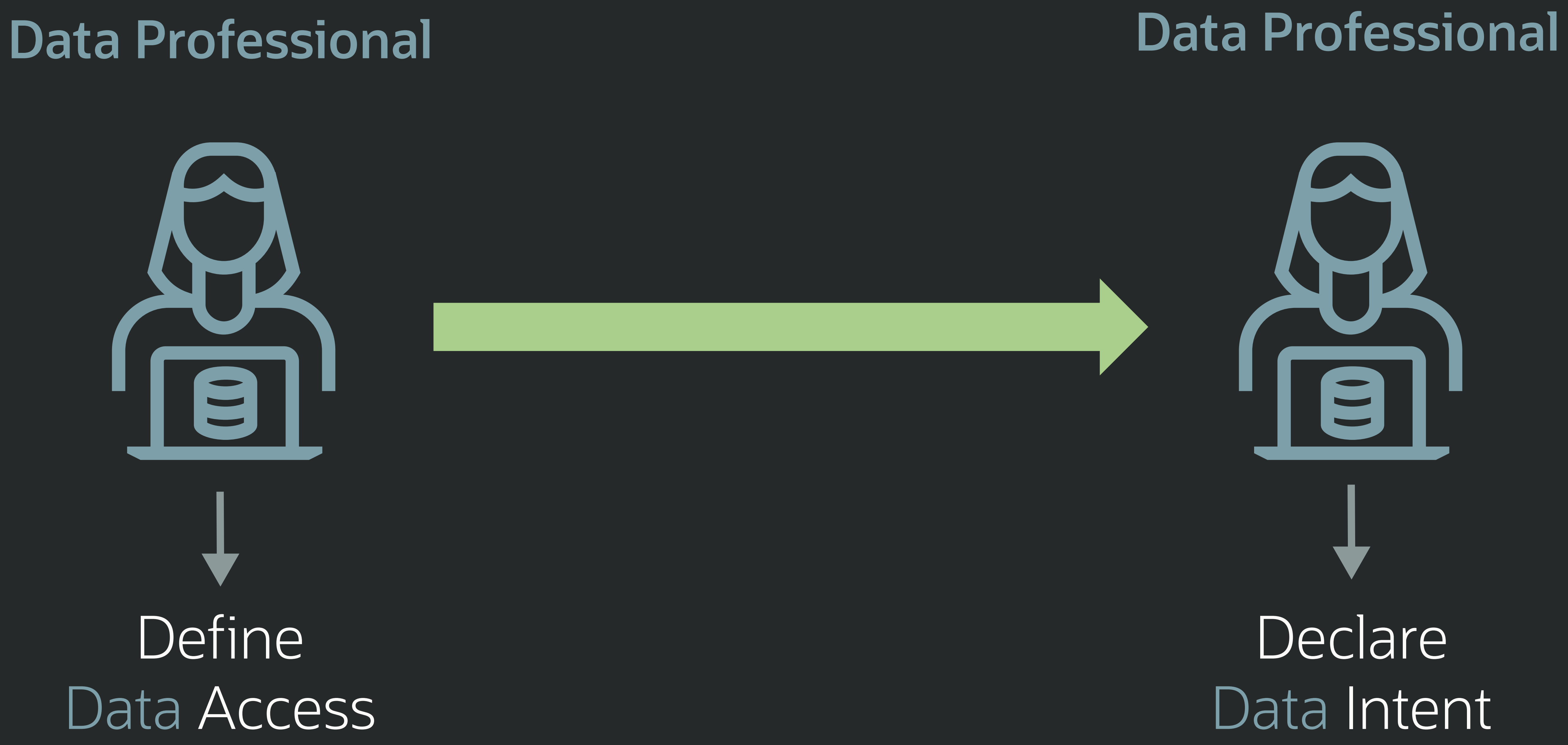
Low-Code Developer

End-User

each face a common challenge

if you need data …

… you must know how to access it



Data Professional

Define
Data Access

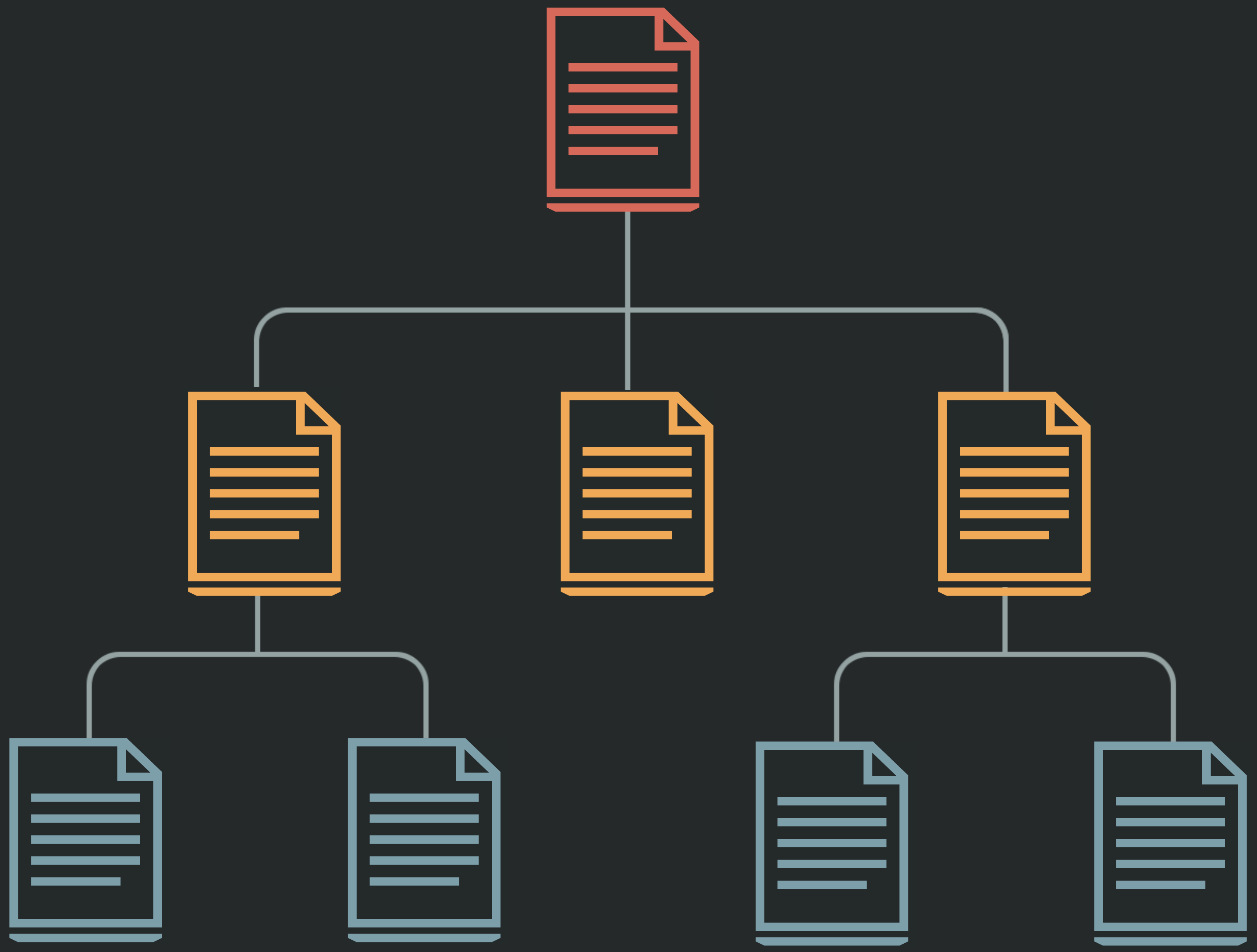
Data Professional

Declare
Data Intent

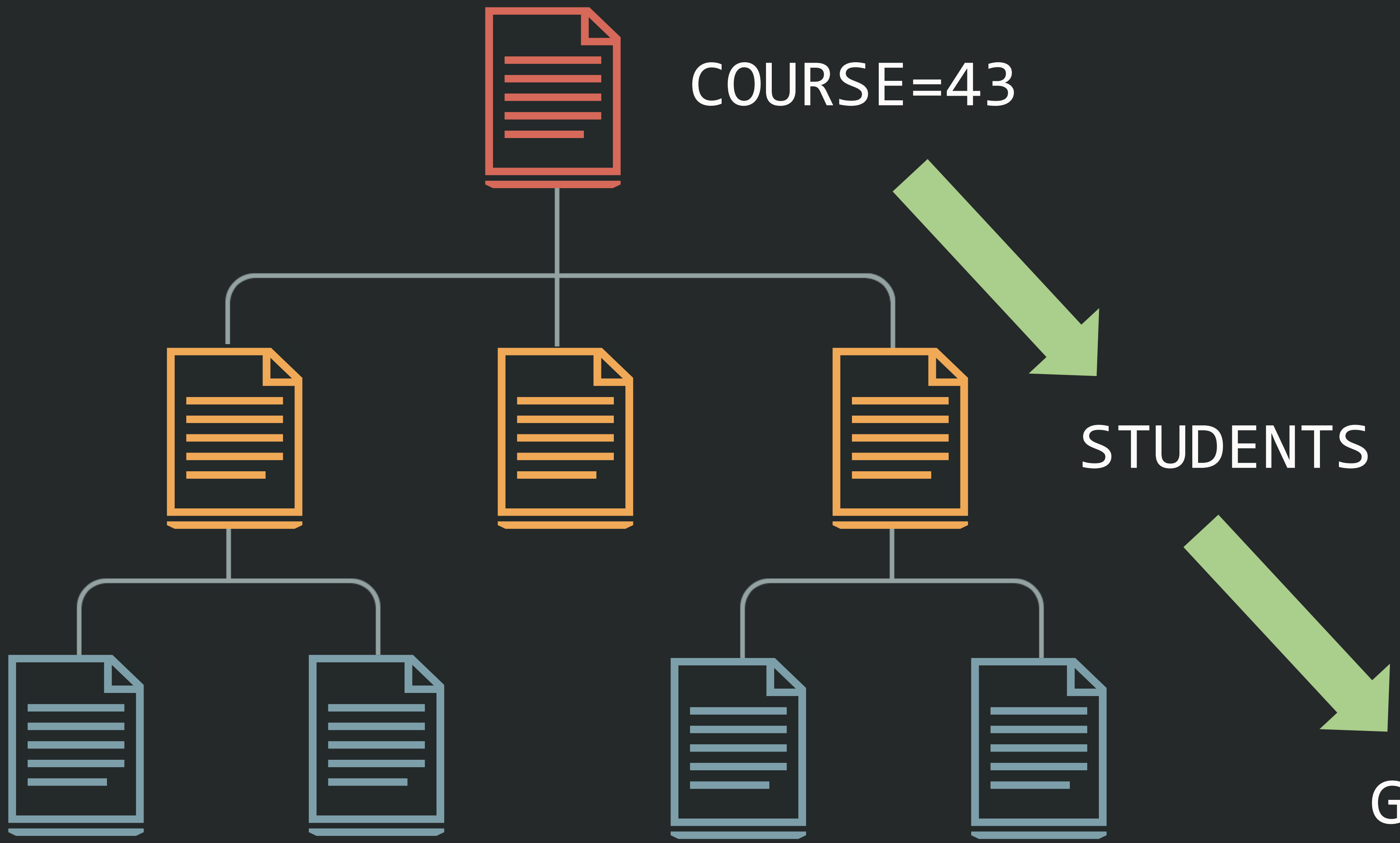
early data professionals were developers

early databases

Hierarchical Databases



hand coded the access



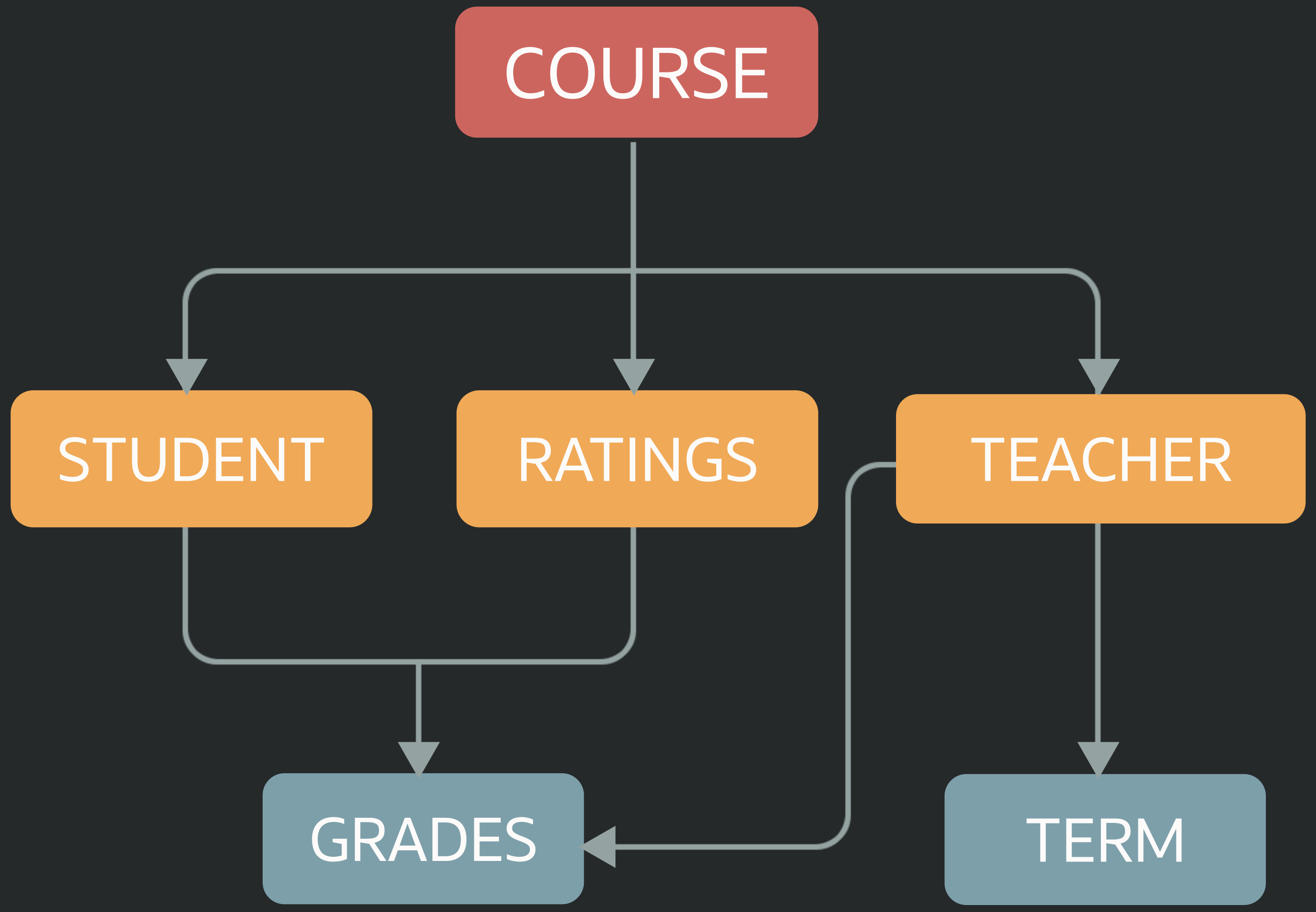
COURSE=43

STUDENTS

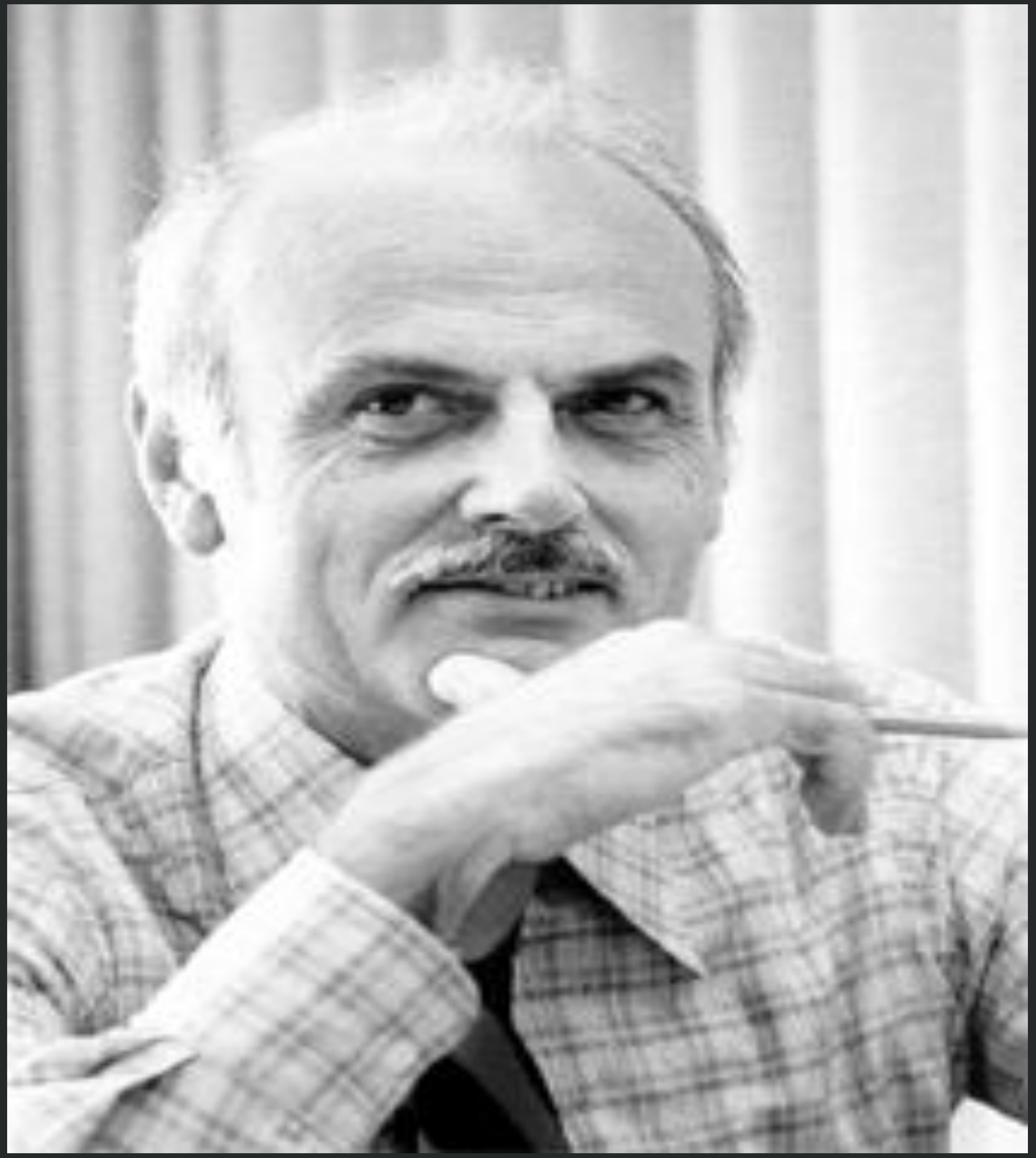
GRADES
(STUDENT=12)

as flexibility demands increased

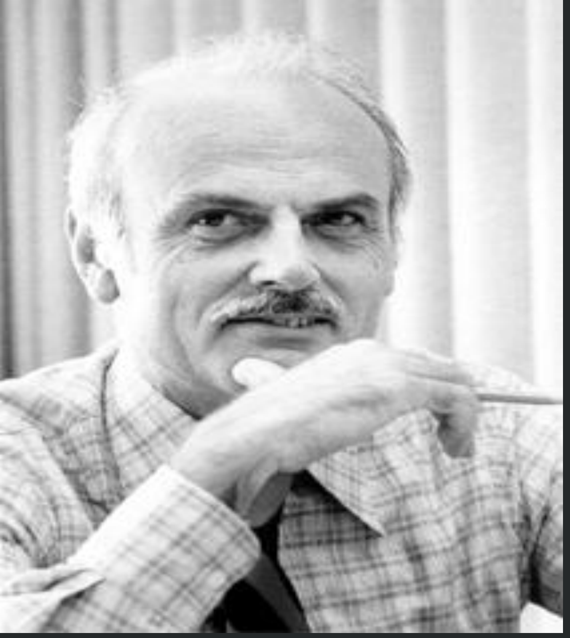
Network Databases



still hand coded the access



Edgar (Ted) Codd



relational model

declarative data intent

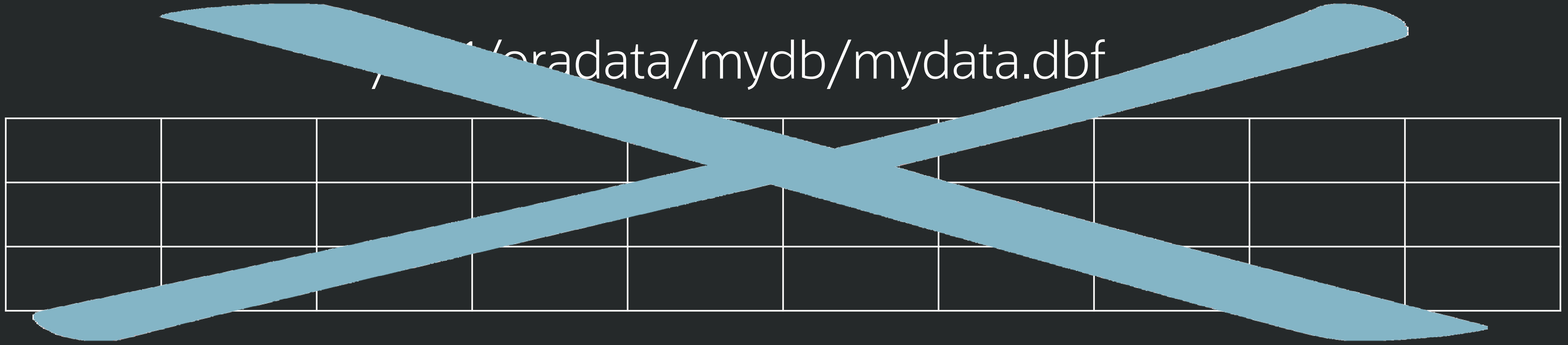
HUGE

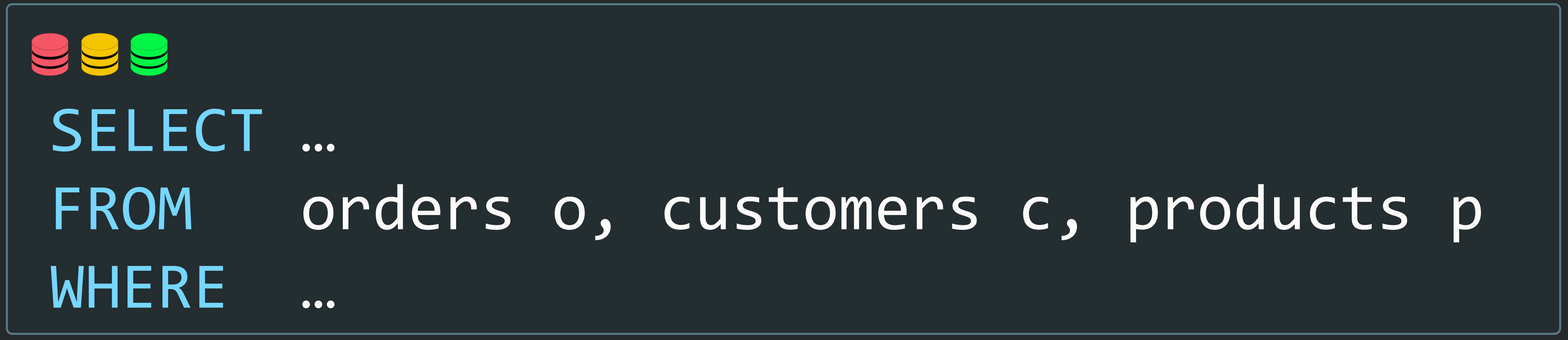
independence from navigation

a) data access by value

independence from structure

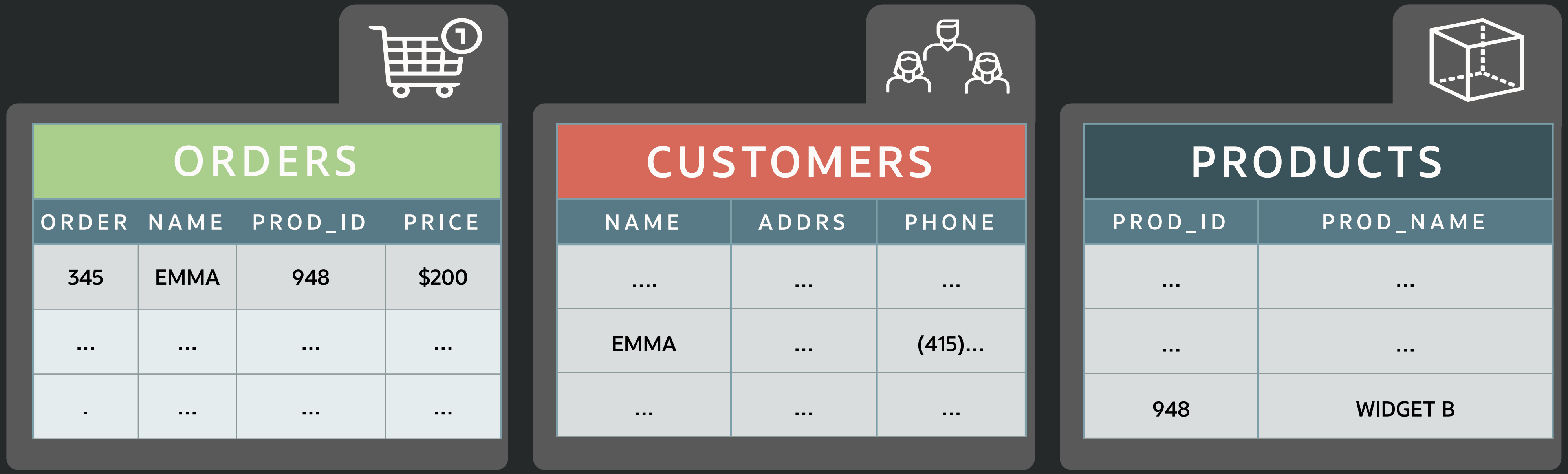
~~/[illegible]1/oradata/mydb/mydata.dbf~~





```
SELECT …
FROM   orders o, customers c, products p
WHERE  …
```

ORDER	NAME	PHONE	PROD_NAME
345	Emma	(415)…	WIDGET B



ORDERS			
ORDER	NAME	PROD_ID	PRICE
345	EMMA	948	$200
…	…	…	…
.	…	…	…

CUSTOMERS		
NAME	ADDRS	PHONE
….	…	…
EMMA	…	(415)…
…	…	…

PRODUCTS	
PROD_ID	PROD_NAME
…	…
…	…
948	WIDGET B

b) data quality

normalisation

(no duplication)

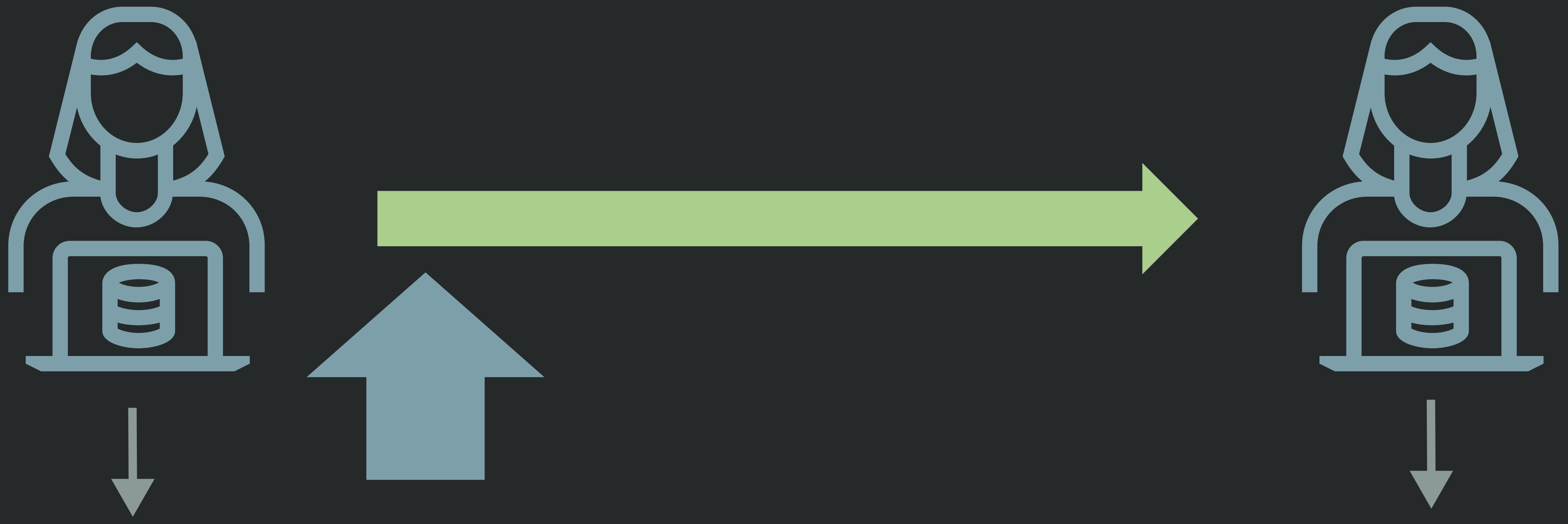
(integrity)

c) data access generated

SQL !

Data Professional

Data Professional



Define
Data Access

Declare
Data Intent

(relational) databases took over the world

Oracle

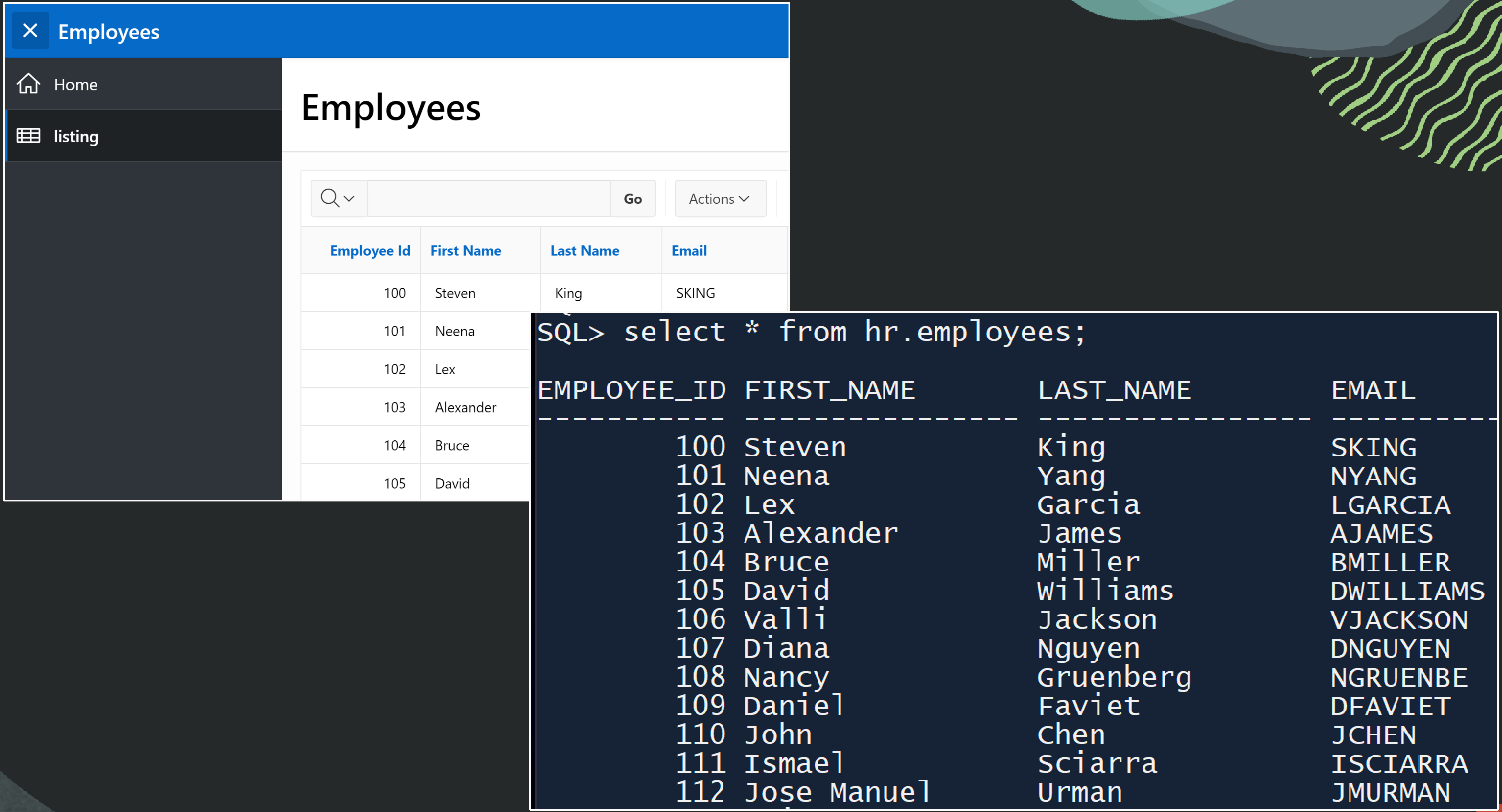
1979

what about developers?

relational

"Oh, you mean use rows and columns!"

fine when app matches the data



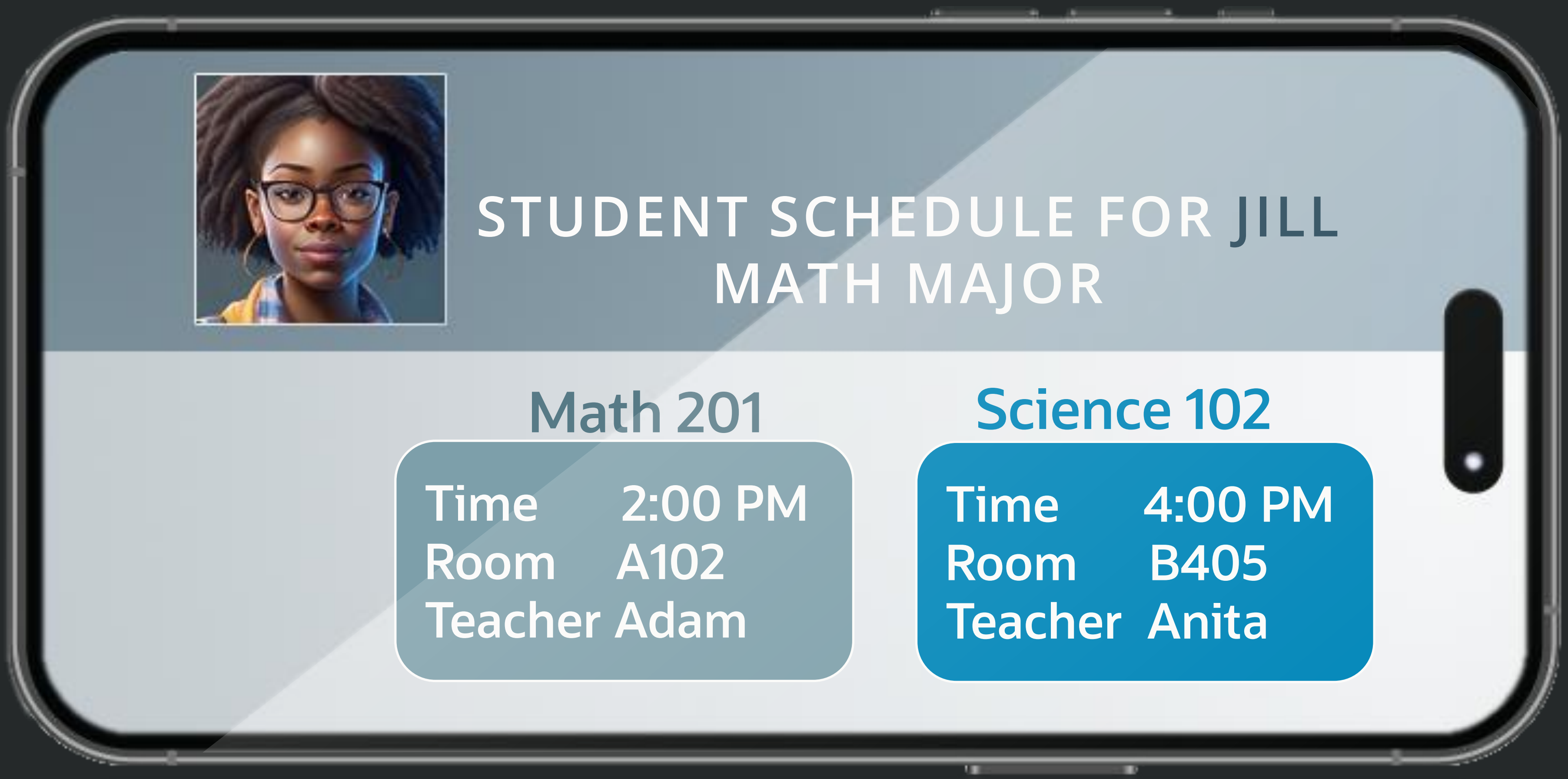
Employees

Employee Id	First Name	Last Name	Email
100	Steven	King	SKING
101	Neena		
102	Lex		
103	Alexander		
104	Bruce		
105	David		

```
SQL> select * from hr.employees;

EMPLOYEE_ID FIRST_NAME           LAST_NAME                 EMAIL
----------- -------------------- ------------------------- ----------
        100 Steven               King                      SKING
        101 Neena                Yang                      NYANG
        102 Lex                  Garcia                    LGARCIA
        103 Alexander            James                     AJAMES
        104 Bruce                Miller                    BMILLER
        105 David                Williams                  DWILLIAMS
        106 Valli                Jackson                   VJACKSON
        107 Diana                Nguyen                    DNGUYEN
        108 Nancy                Gruenberg                 NGRUENBE
        109 Daniel               Faviet                    DFAVIET
        110 John                 Chen                      JCHEN
        111 Ismael               Sciarra                   ISCIARRA
        112 Jose Manuel          Urman                     JMURMAN
```

but when it doesn't?

hypothetical college app



STUDENT SCHEDULE FOR JILL
MATH MAJOR

Math 201

Time	2:00 PM
Room	A102
Teacher	Adam

Science 102

Time	4:00 PM
Room	B405
Teacher	Anita

data model

STUDENT

STUID	SNAME	MAJOR	YEAR
S3245	Jill	Math	First
...	...	...	...
...	...	...	...
...	...	...	...

COURSE

CID	CLASS	ROOM	TIME	TCHID
C123	MATH 201	A102	14:00	T543
C345	SCIENCE 102	B405	16:00	T789
...	...	...	...	...
...	...	...	...	...

STUDENT COURSES

STUID	CID
S3245	C123
...	...
S3245	C345
...	...

TEACHER

TCHID	TEACHER	TINFO
...	...	...
T543	Adam	...
T789	Anita	...
...	...	...

storage independence

data integrity/consistency

awesome but ...

declarative SQL

adhoc query

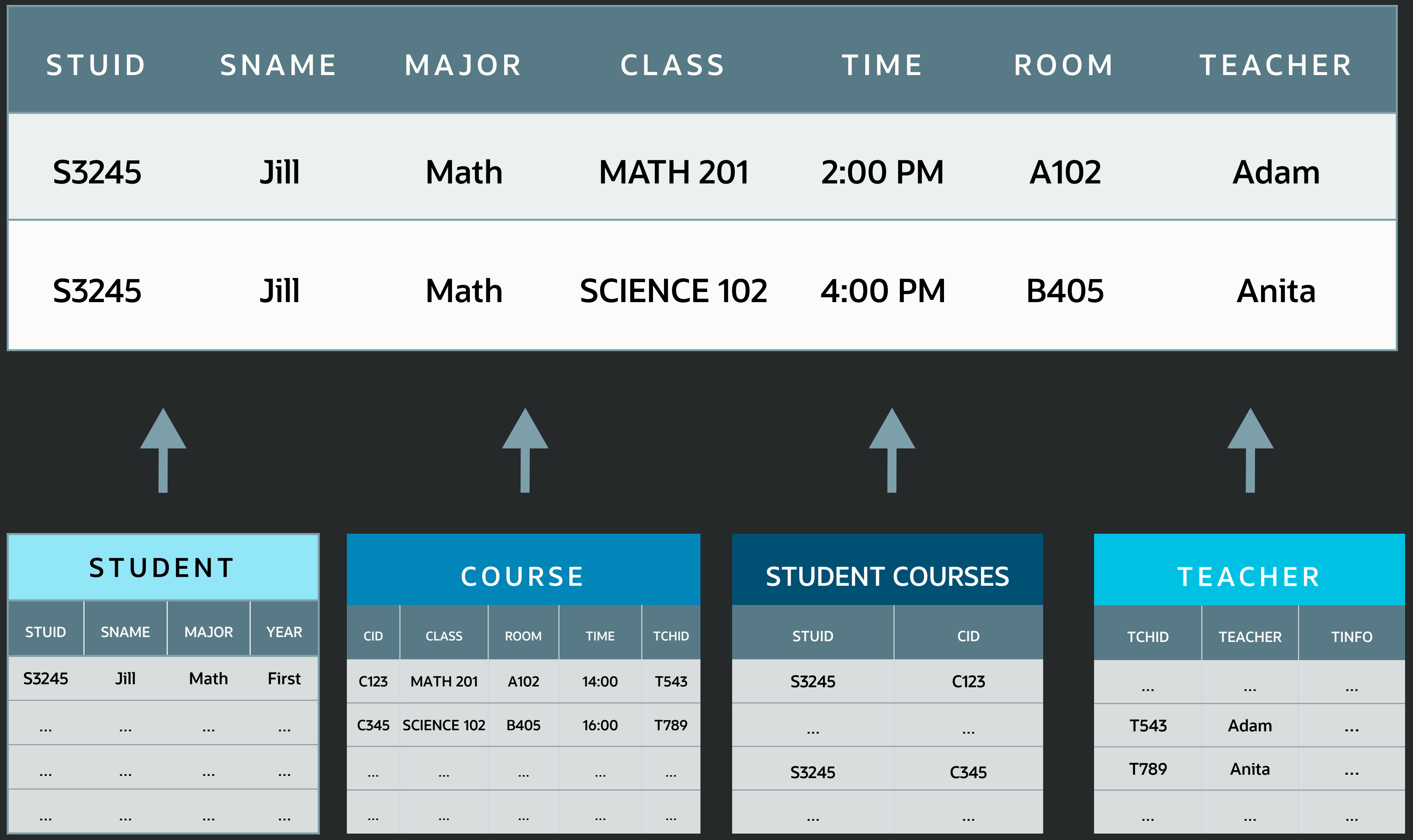
modern code is not relational

modern code is OO

modern code is hierarchical

Student.java

```
public Student(String name,
                String major,
                Schedule sched
                )
 {
   this.studentID = name;
   this.major = age;
   this.schedule = sched;
 }

public Schedule(String sched_time,
                 String course,
                 String room,
                 String teacher)
 {
   this.time = sched_time;
   this.course = course;
   this.room = room;
   this.teacher = teacher;
 }
```



STUID	SNAME	MAJOR	CLASS	TIME	ROOM	TEACHER
S3245	Jill	Math	MATH 201	2:00 PM	A102	Adam
S3245	Jill	Math	SCIENCE 102	4:00 PM	B405	Anita

STUDENT

STUID	SNAME	MAJOR	YEAR
S3245	Jill	Math	First
...	...	...	...
...	...	...	...
...	...	...	...

COURSE

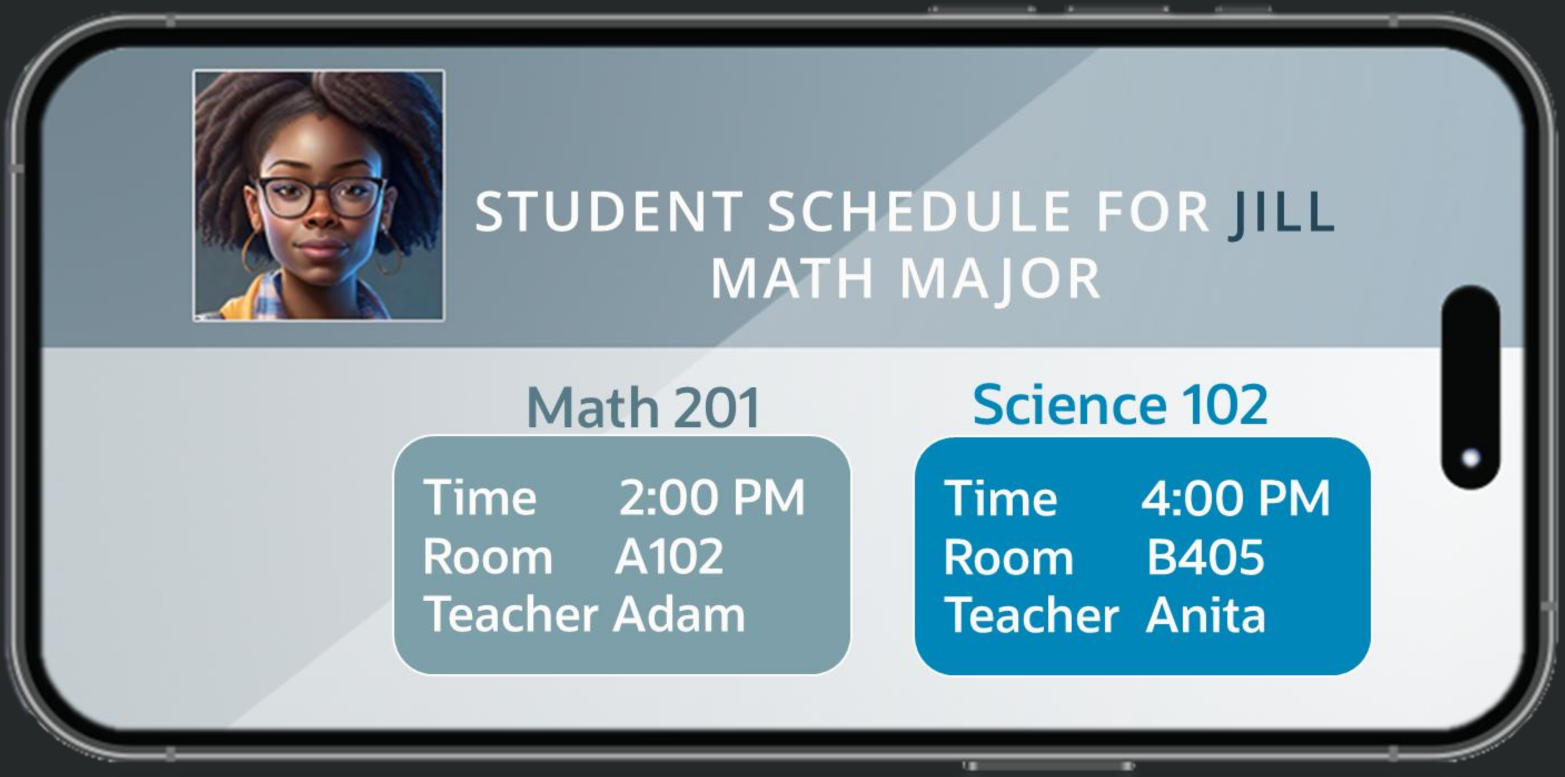
CID	CLASS	ROOM	TIME	TCHID
C123	MATH 201	A102	14:00	T543
C345	SCIENCE 102	B405	16:00	T789
...	...	...	...	...
...	...	...	...	...

STUDENT COURSES

STUID	CID
S3245	C123
...	...
S3245	C345
...	...

TEACHER

TCHID	TEACHER	TINFO
...	...	...
T543	Adam	...
T789	Anita	...
...	...	...



STUDENT SCHEDULE FOR JILL
MATH MAJOR

Math 201

Time 2:00 PM
Room A102
Teacher Adam

Science 102

Time 4:00 PM
Room B405
Teacher Anita

STUID	SNAME	MAJOR	CLASS	TIME	ROOM	TEACHER
S3245	Jill	Math	MATH 201	2:00 PM	A102	Adam
S3245	Jill	Math	SCIENCE 102	4:00 PM	B405	Anita

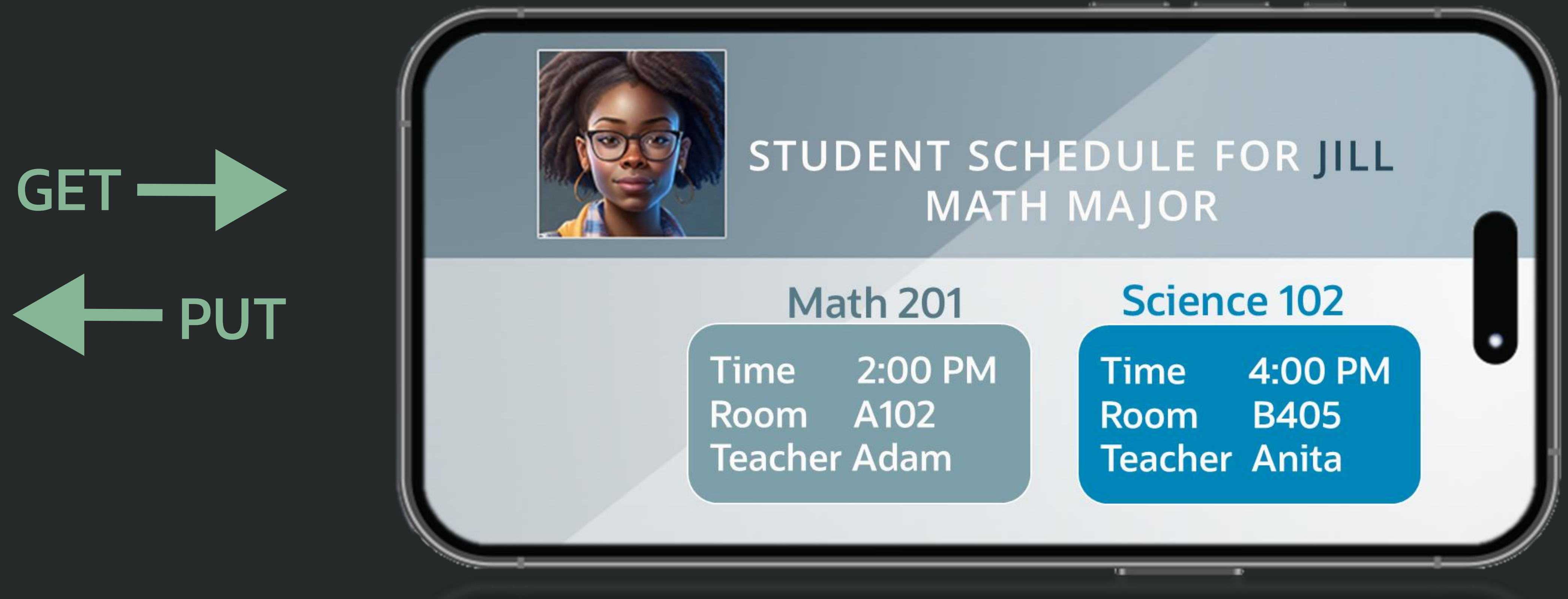
Repeated student data

data mapping = more code/complexity

hierarchical code needs hierarchical data

SCHEDULE FOR: JILL

```
{
 "studentID"    : "S3245",
 "name"         : "Jill",
 "major"        : "Math",
 "schedule"     :
   [  {
        "time"    : "14:00",
        "course"  : "Math 201",
        "room"    : "A102",
        "teacher" : "Adam"
      },
      {
        "time"    : "16:00",
        "course"  : "Science 102",
        "room"    : "B405",
        "teacher" : "Anita"
      }
   ]
}
```



STUDENT SCHEDULE FOR JILL
MATH MAJOR

Math 201
Time 2:00 PM
Room A102
Teacher Adam

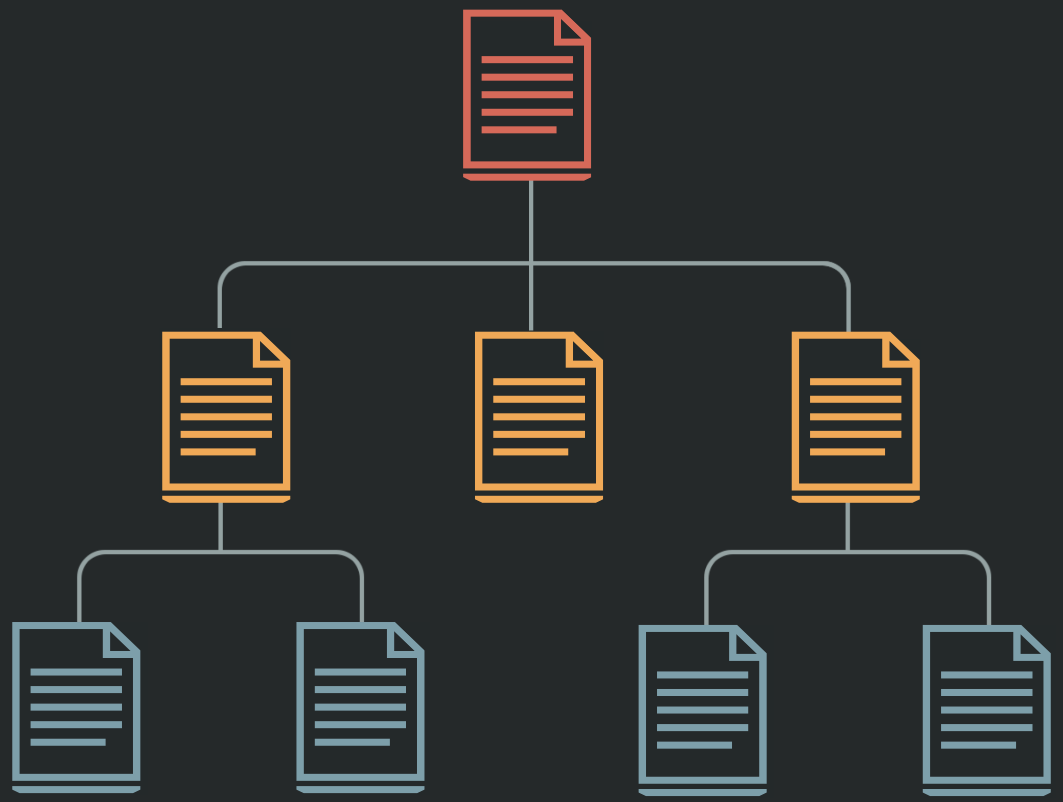
Science 102
Time 4:00 PM
Room B405
Teacher Anita

relational ?

Document and Key-Value Databases ("NoSQL")

↑

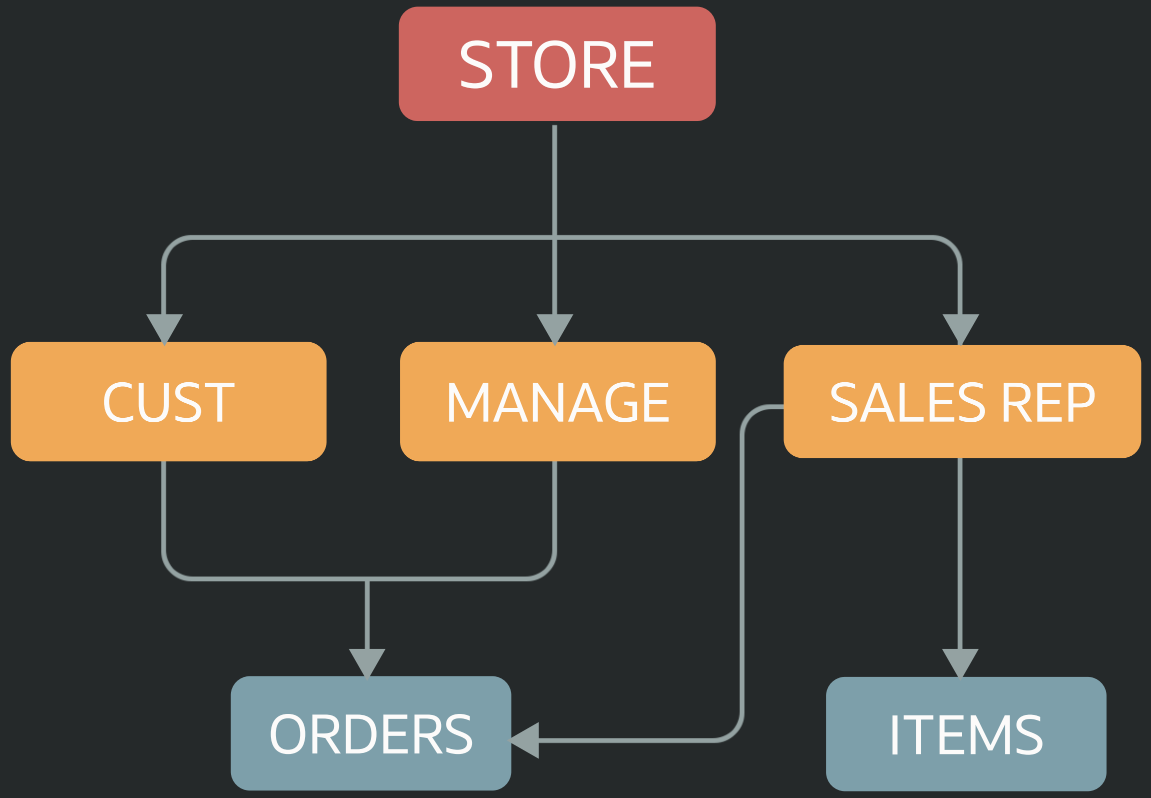
Hierarchical Databases



Graph Databases

↑

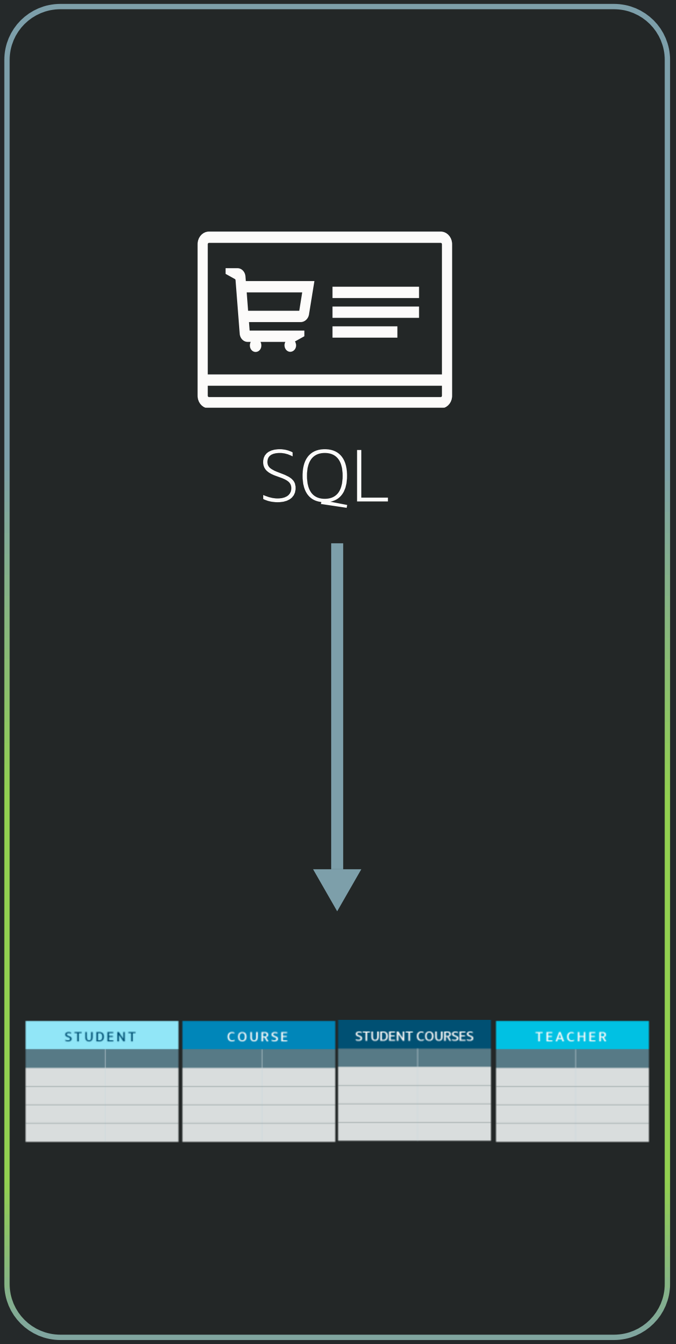
Network Databases

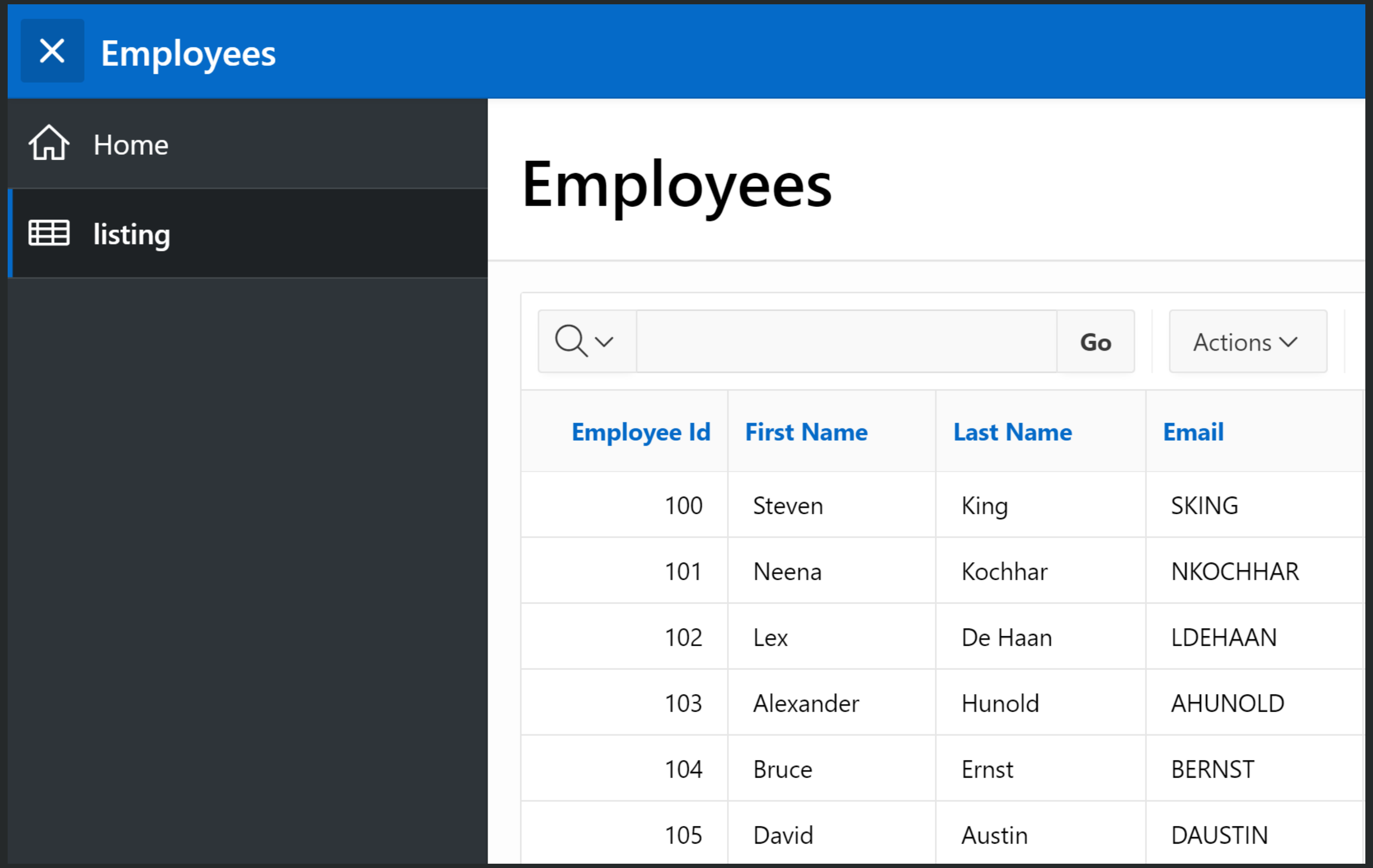


all have a common theme

data usage d f e data storage

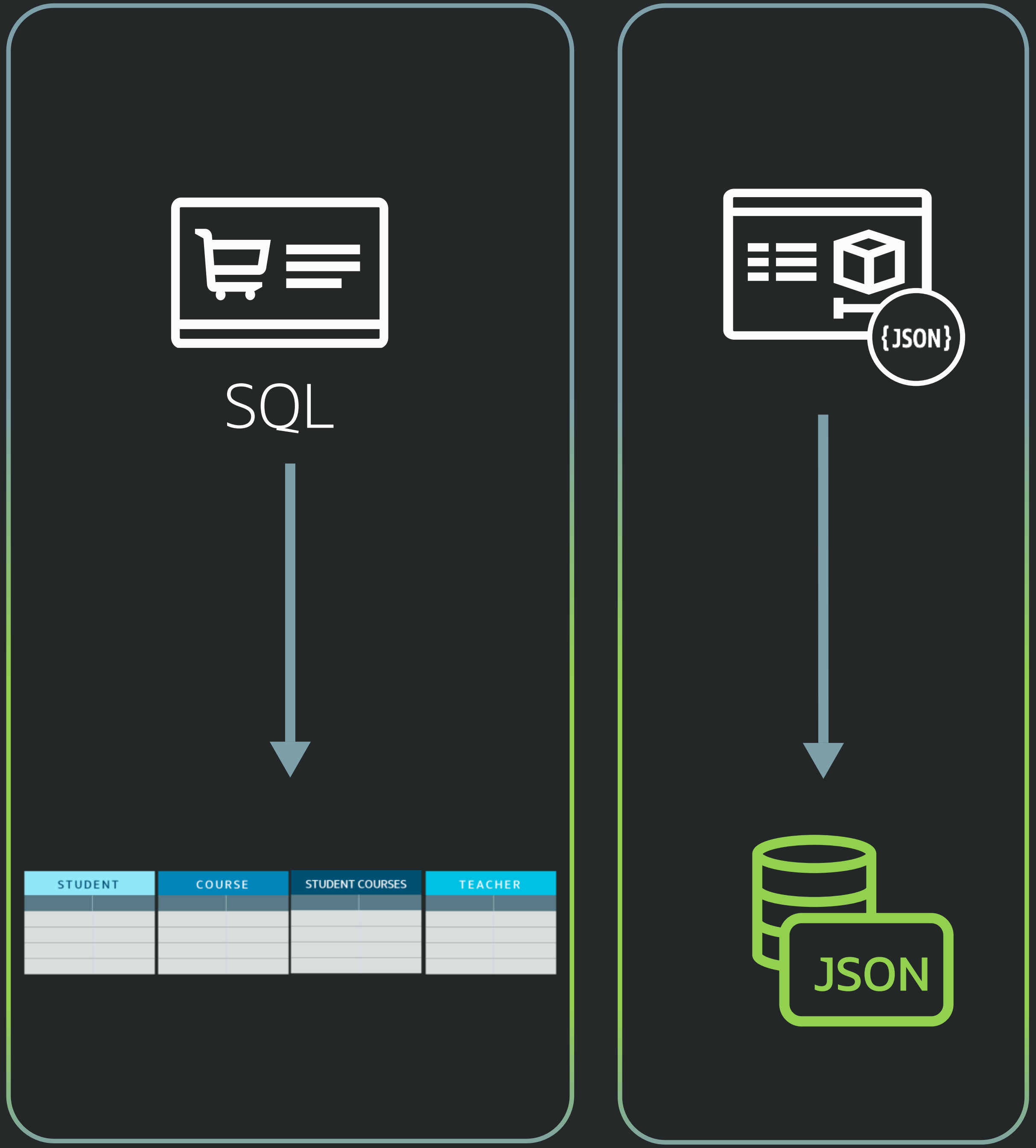
a complete failure of the IT profession

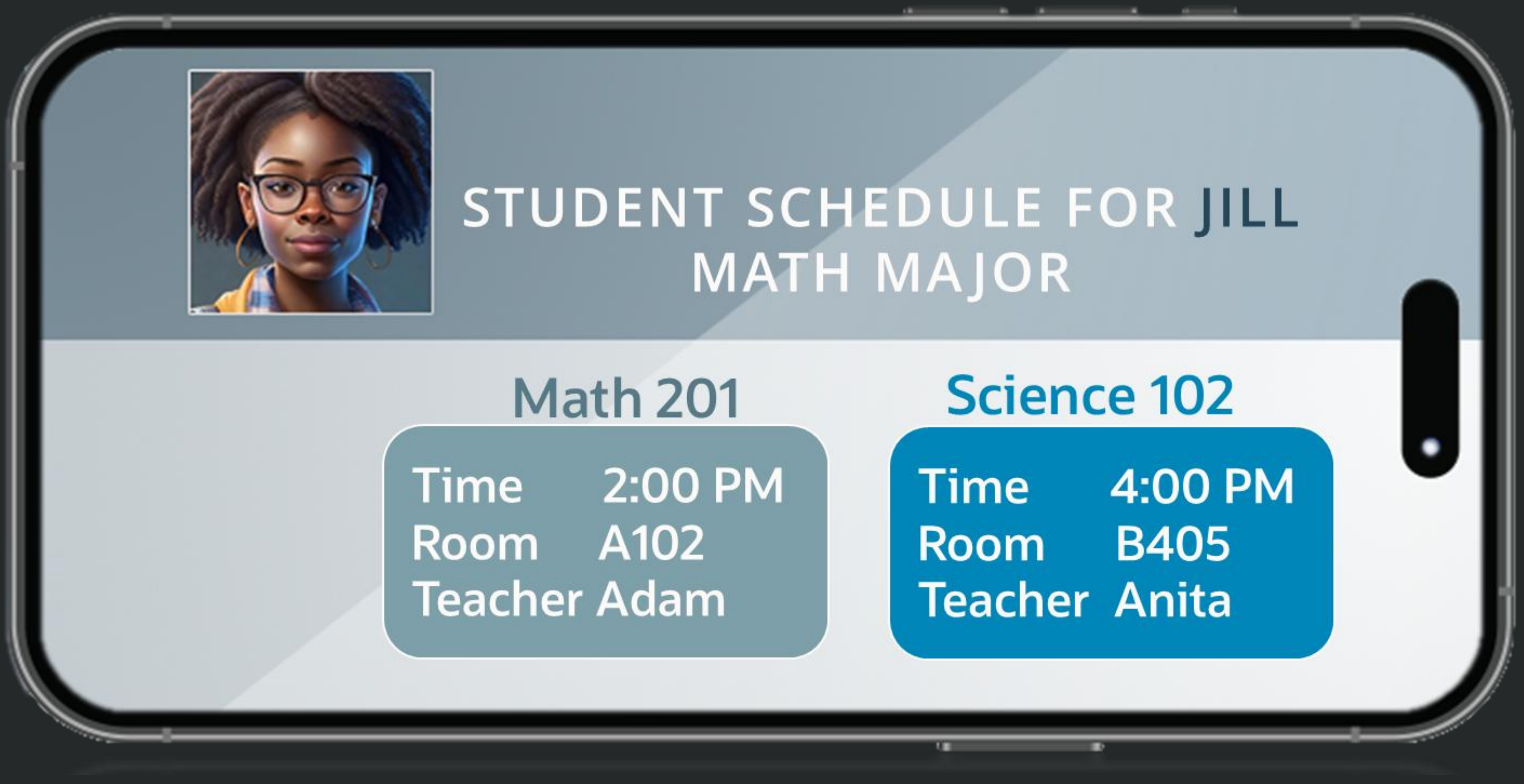




Employees

Employee Id	First Name	Last Name	Email
100	Steven	King	SKING
101	Neena	Kochhar	NKOCHHAR
102	Lex	De Haan	LDEHAAN
103	Alexander	Hunold	AHUNOLD
104	Bruce	Ernst	BERNST
105	David	Austin	DAUSTIN





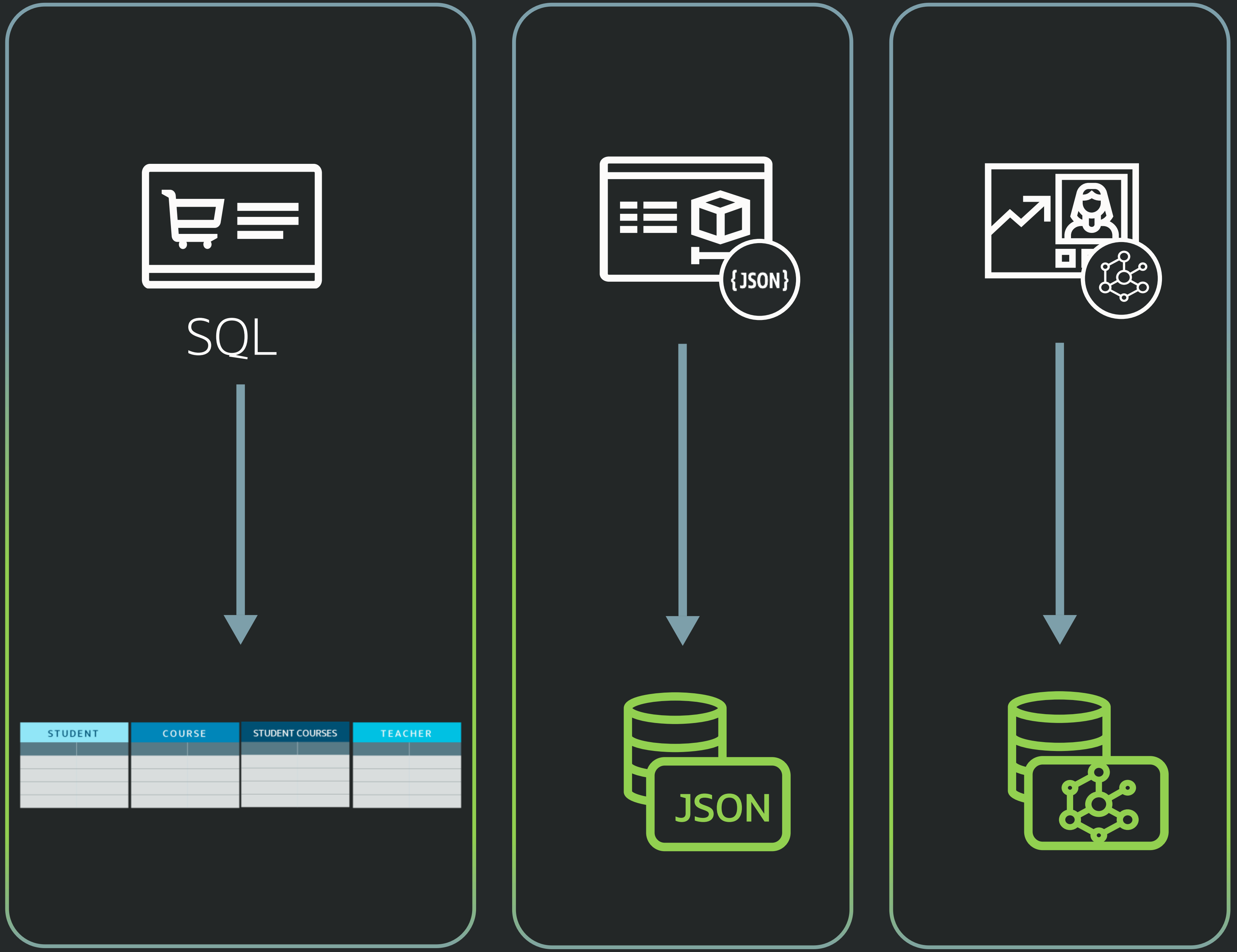
STUDENT SCHEDULE FOR JILL
MATH MAJOR

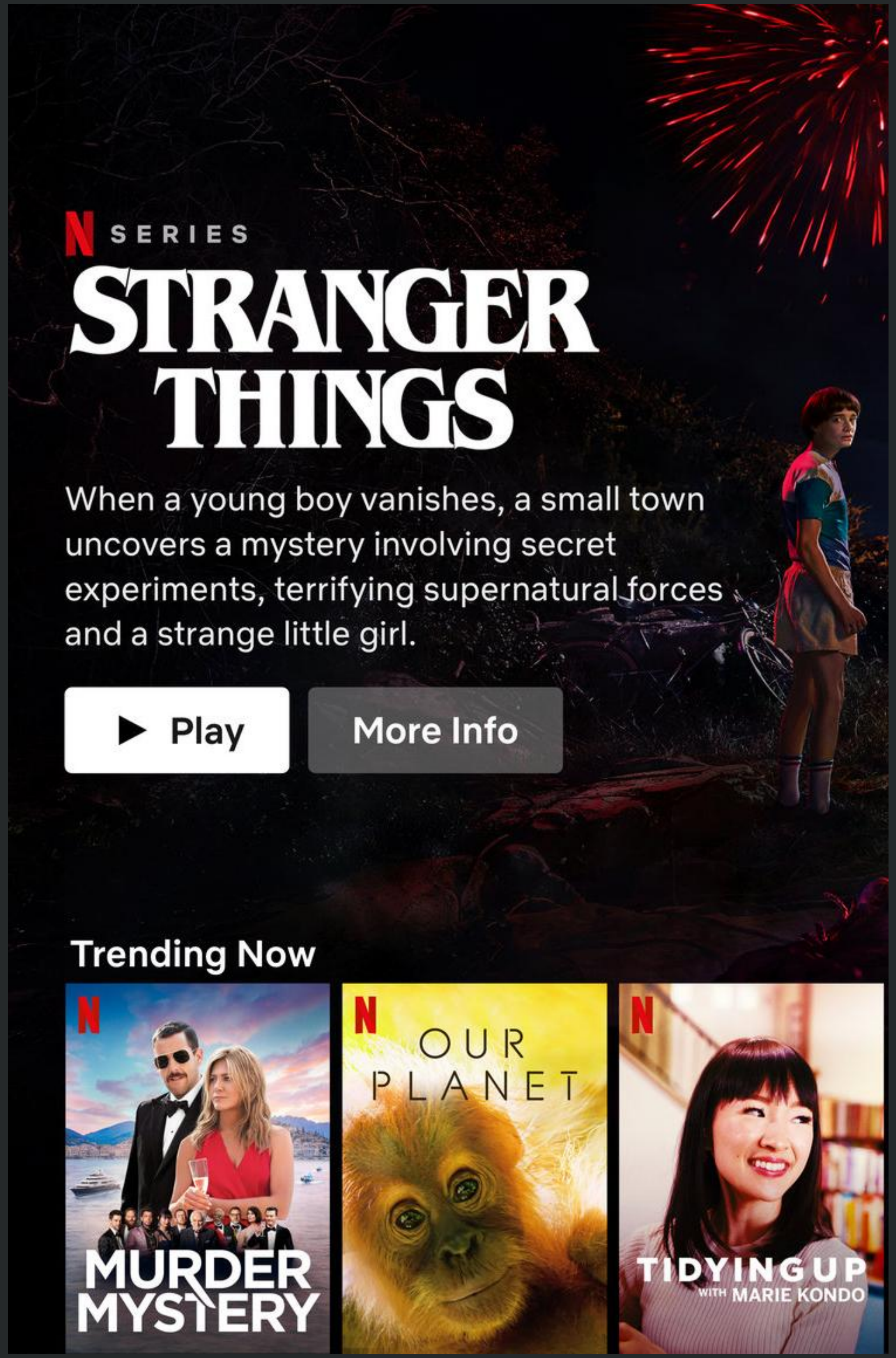
Math 201

Time 2:00 PM
Room A102
Teacher Adam

Science 102

Time 4:00 PM
Room B405
Teacher Anita





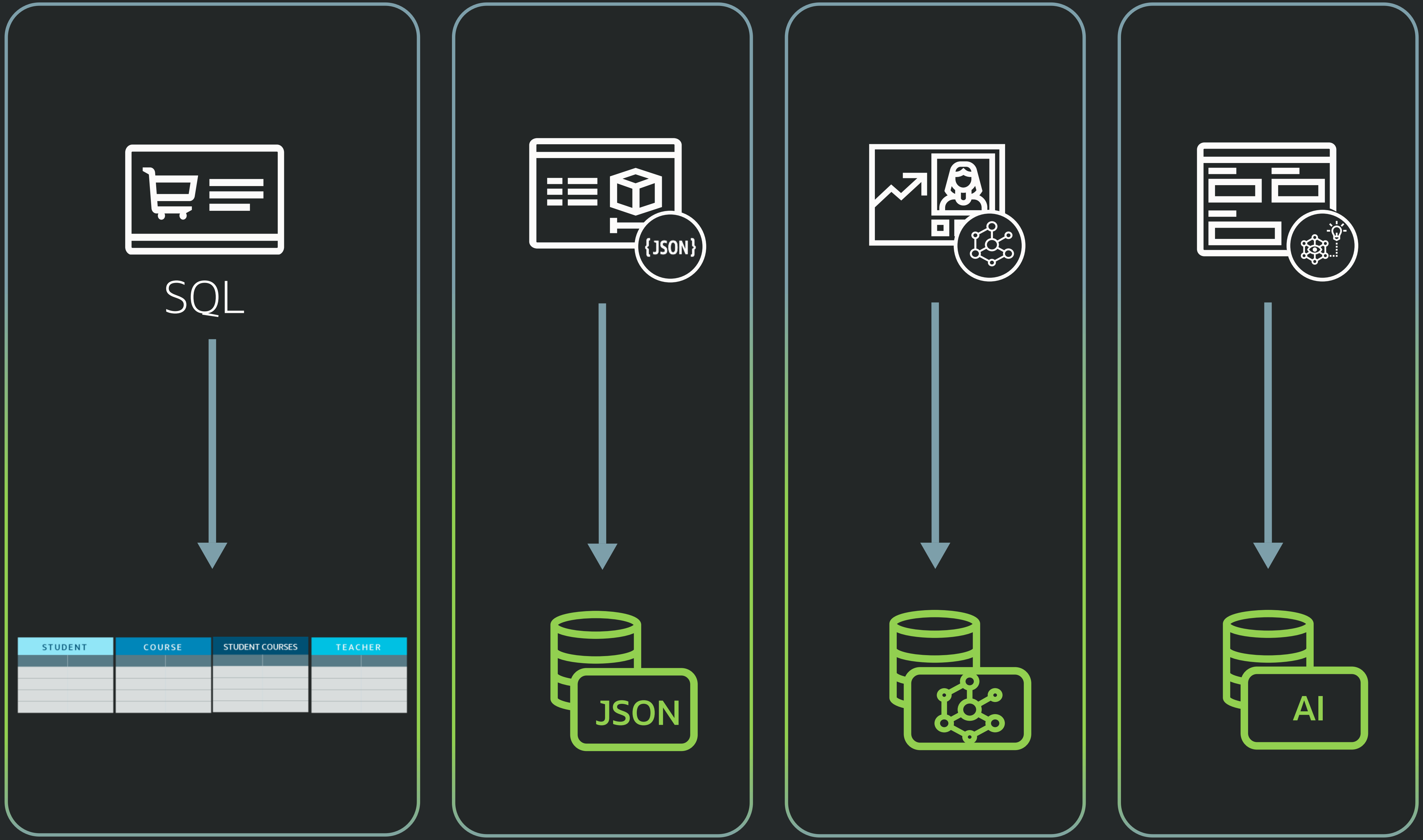
N SERIES

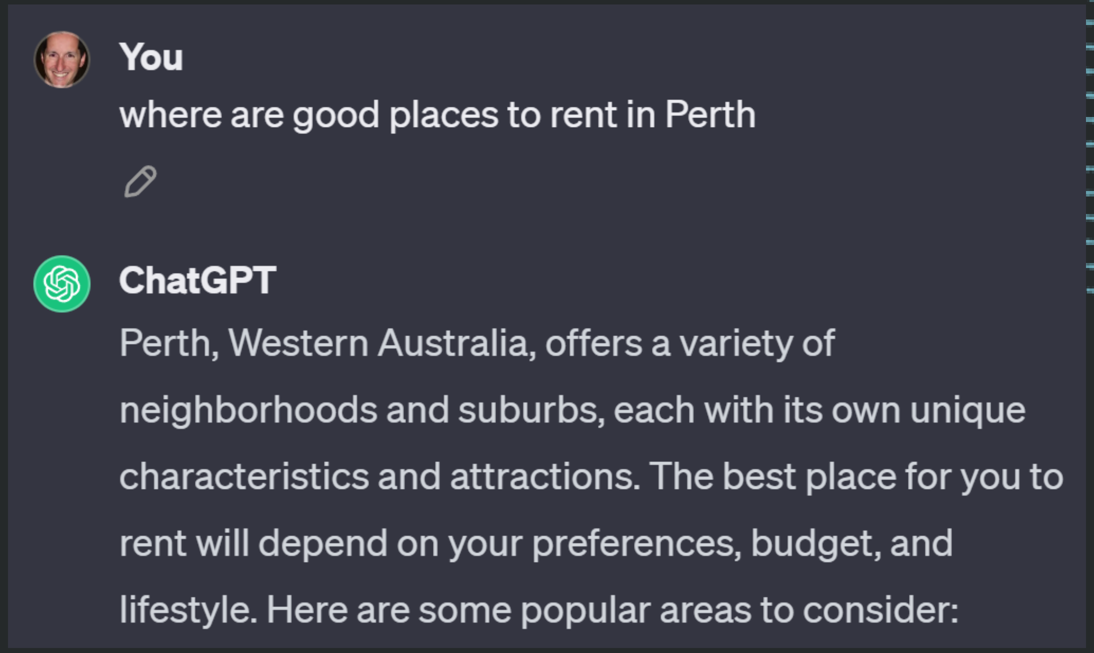
STRANGER THINGS

When a young boy vanishes, a small town uncovers a mystery involving secret experiments, terrifying supernatural forces and a strange little girl.

▶ Play | More Info

Trending Now





You

where are good places to rent in Perth

ChatGPT

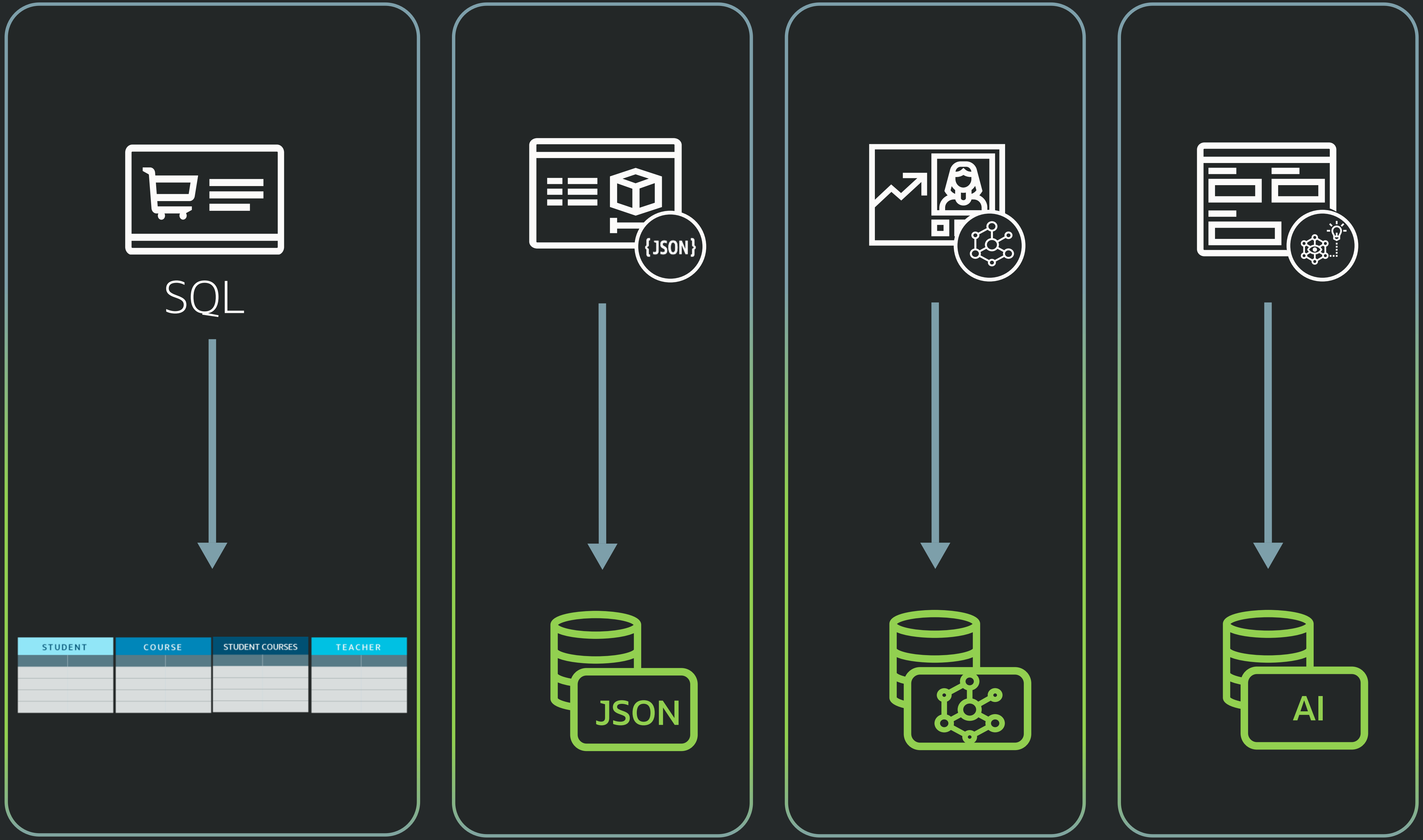
Perth, Western Australia, offers a variety of neighborhoods and suburbs, each with its own unique characteristics and attractions. The best place for you to rent will depend on your preferences, budget, and lifestyle. Here are some popular areas to consider:

it sounds good but …

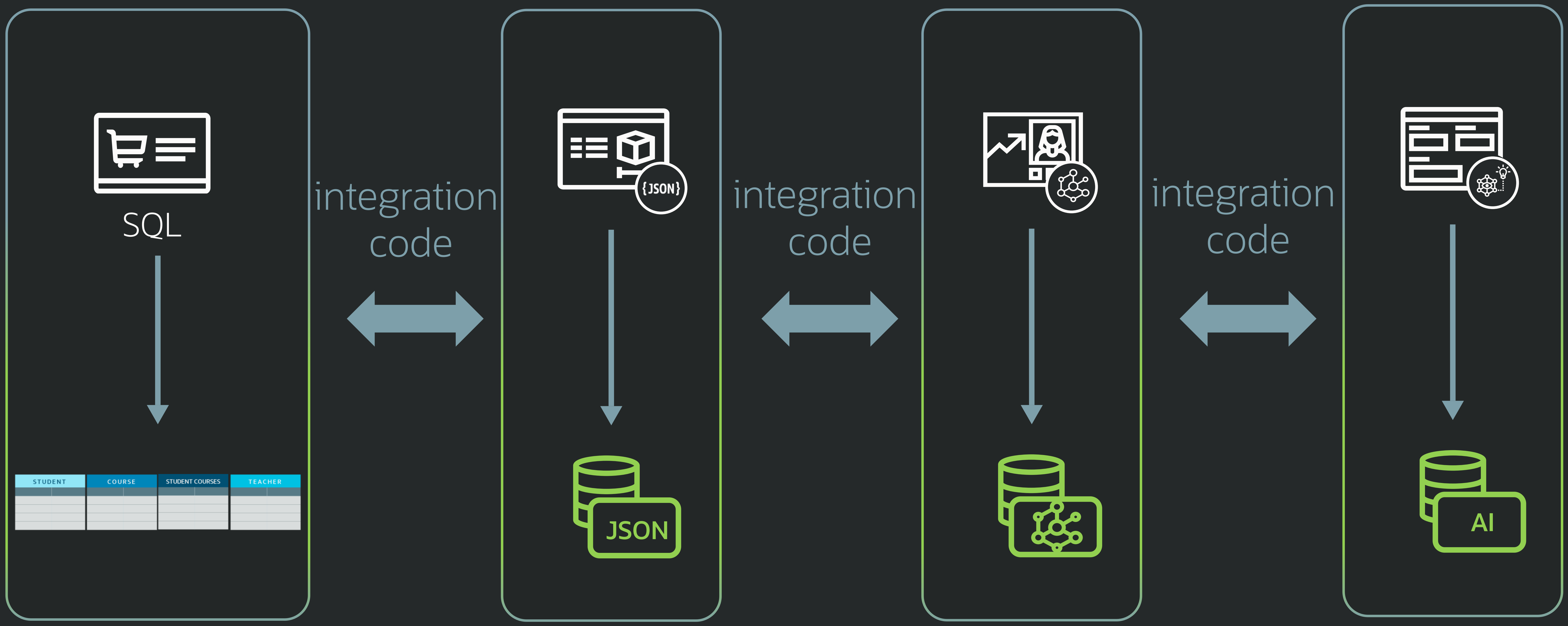
it got worse for customers

"I swear if I get another version of the same employee record, we are FINISHED!"

it got worse for developers



not the reality



SQL

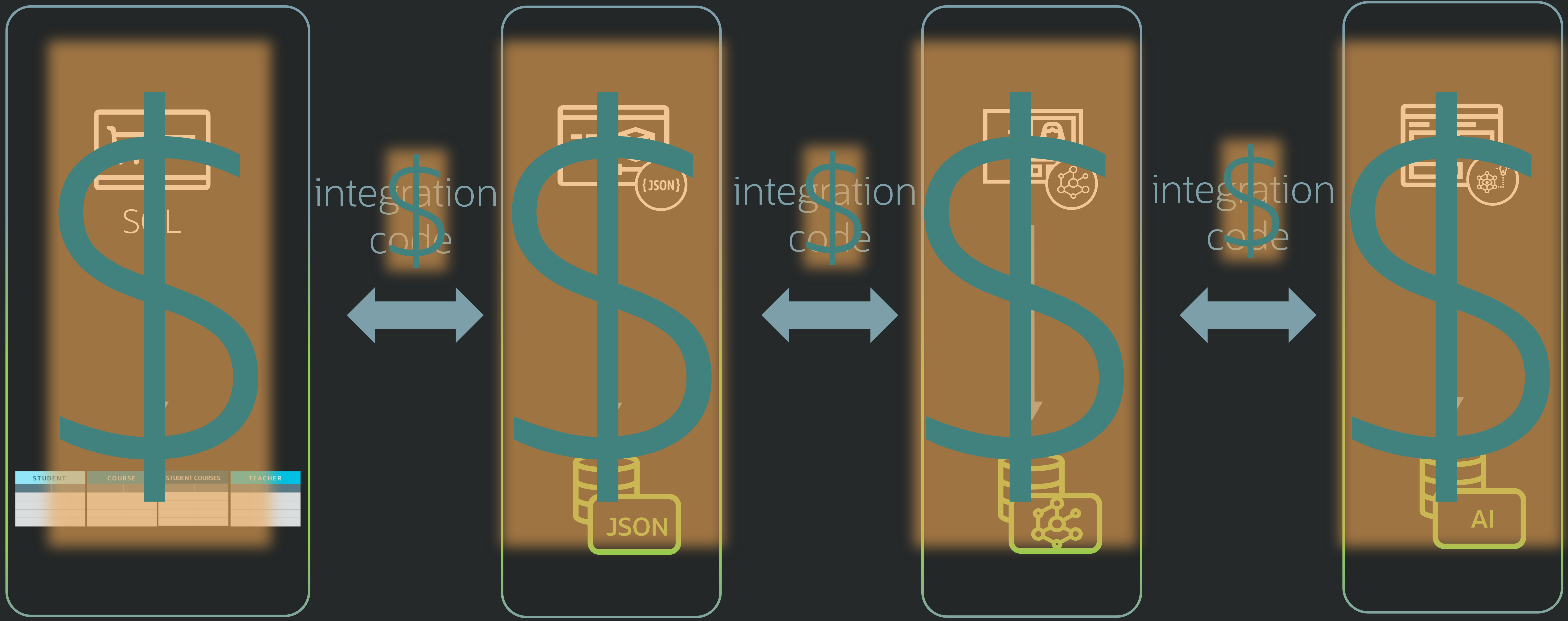
integration code

integration code

integration code

it's great for

☺



SQL

integration code

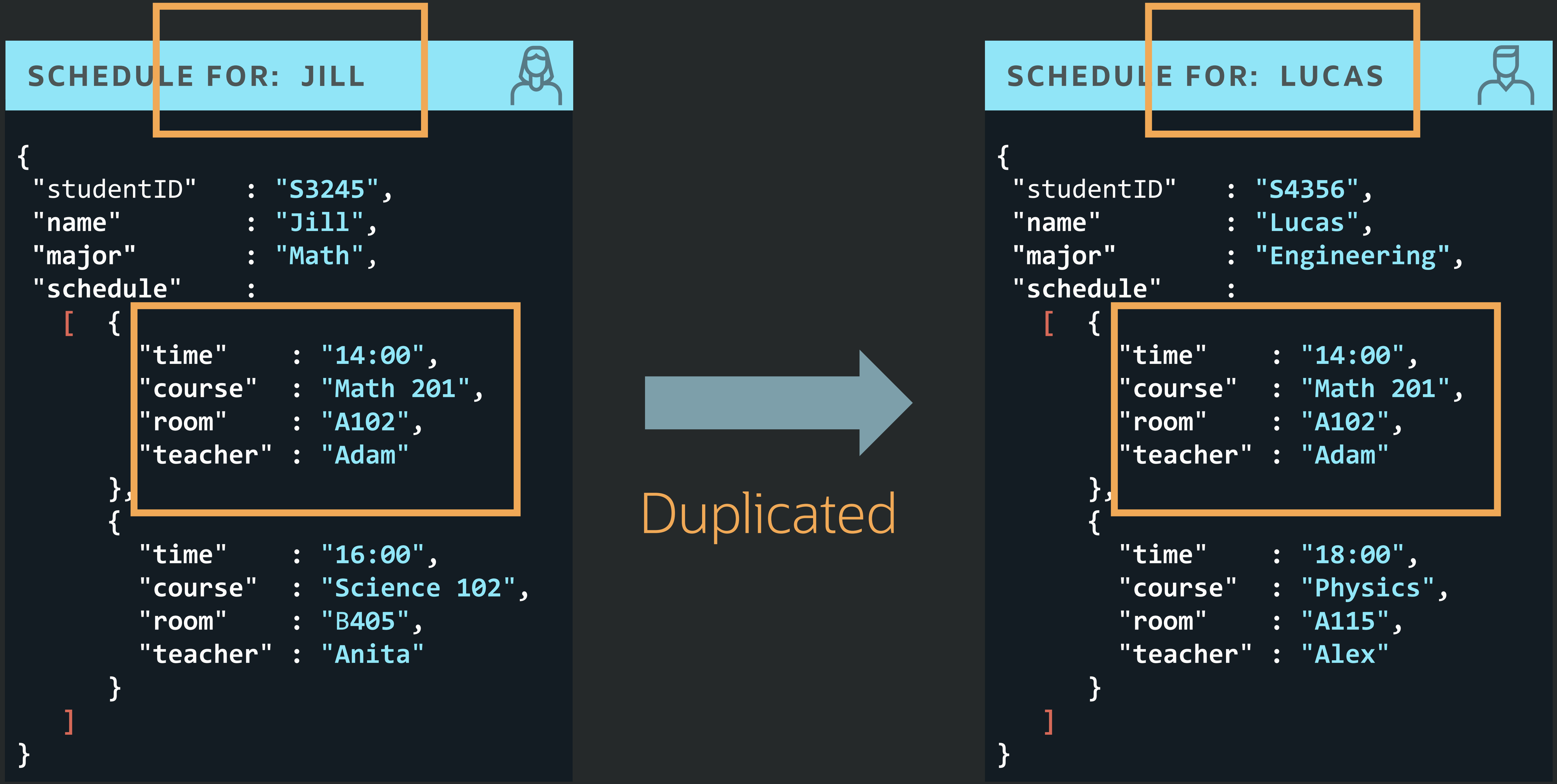
integration code

integration code

the data quality issues came back

relational = declarative data accuracy

its why we moved away from hierarchy



SCHEDULE FOR: JILL

```
{
 "studentID"   : "S3245",
 "name"        : "Jill",
 "major"       : "Math",
 "schedule"    :
   [  {
        "time"    : "14:00",
        "course"  : "Math 201",
        "room"    : "A102",
        "teacher" : "Adam"
      },
      {
        "time"    : "16:00",
        "course"  : "Science 102",
        "room"    : "B405",
        "teacher" : "Anita"
      }
   ]
}
```

Duplicated

SCHEDULE FOR: LUCAS

```
{
 "studentID"   : "S4356",
 "name"        : "Lucas",
 "major"       : "Engineering",
 "schedule"    :
   [  {
        "time"    : "14:00",
        "course"  : "Math 201",
        "room"    : "A102",
        "teacher" : "Adam"
      },
      {
        "time"    : "18:00",
        "course"  : "Physics",
        "room"    : "A115",
        "teacher" : "Alex"
      }
   ]
}
```

our big idea

data usage

↕

data storage

data usage by intent

not by storage

"I need documents"

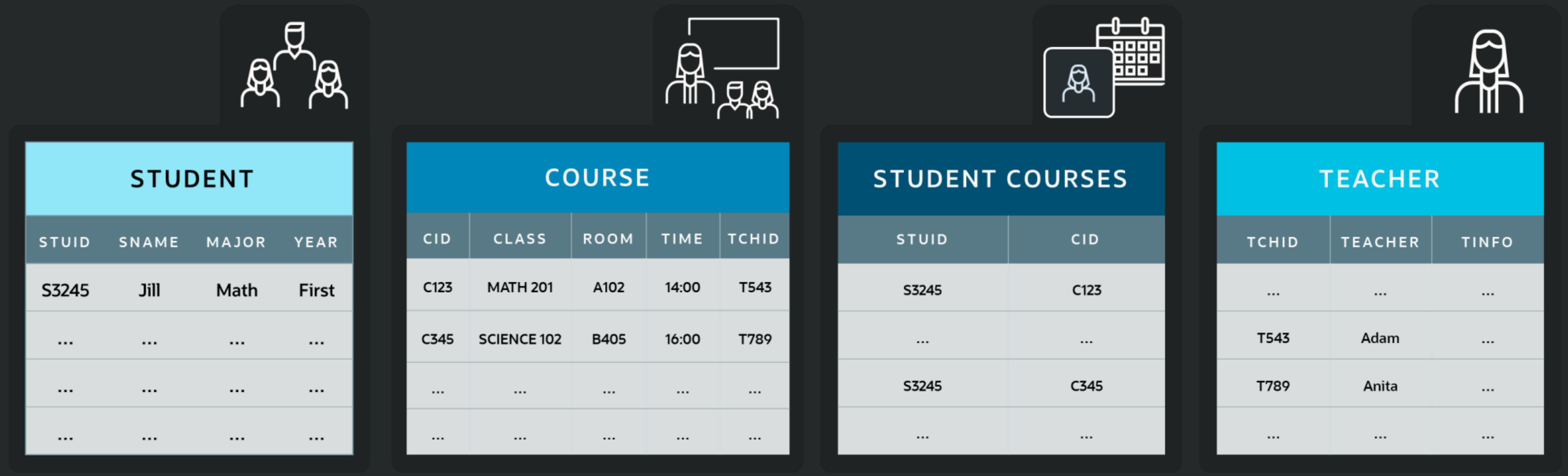
{JSON}

"I need graph"

"I need rows"

single truth

we know how to do single truth



STUDENT

STUID	SNAME	MAJOR	YEAR
S3245	Jill	Math	First
...	...	...	...
...	...	...	...
...	...	...	...

COURSE

CID	CLASS	ROOM	TIME	TCHID
C123	MATH 201	A102	14:00	T543
C345	SCIENCE 102	B405	16:00	T789
...	...	...	...	...
...	...	...	...	...

STUDENT COURSES

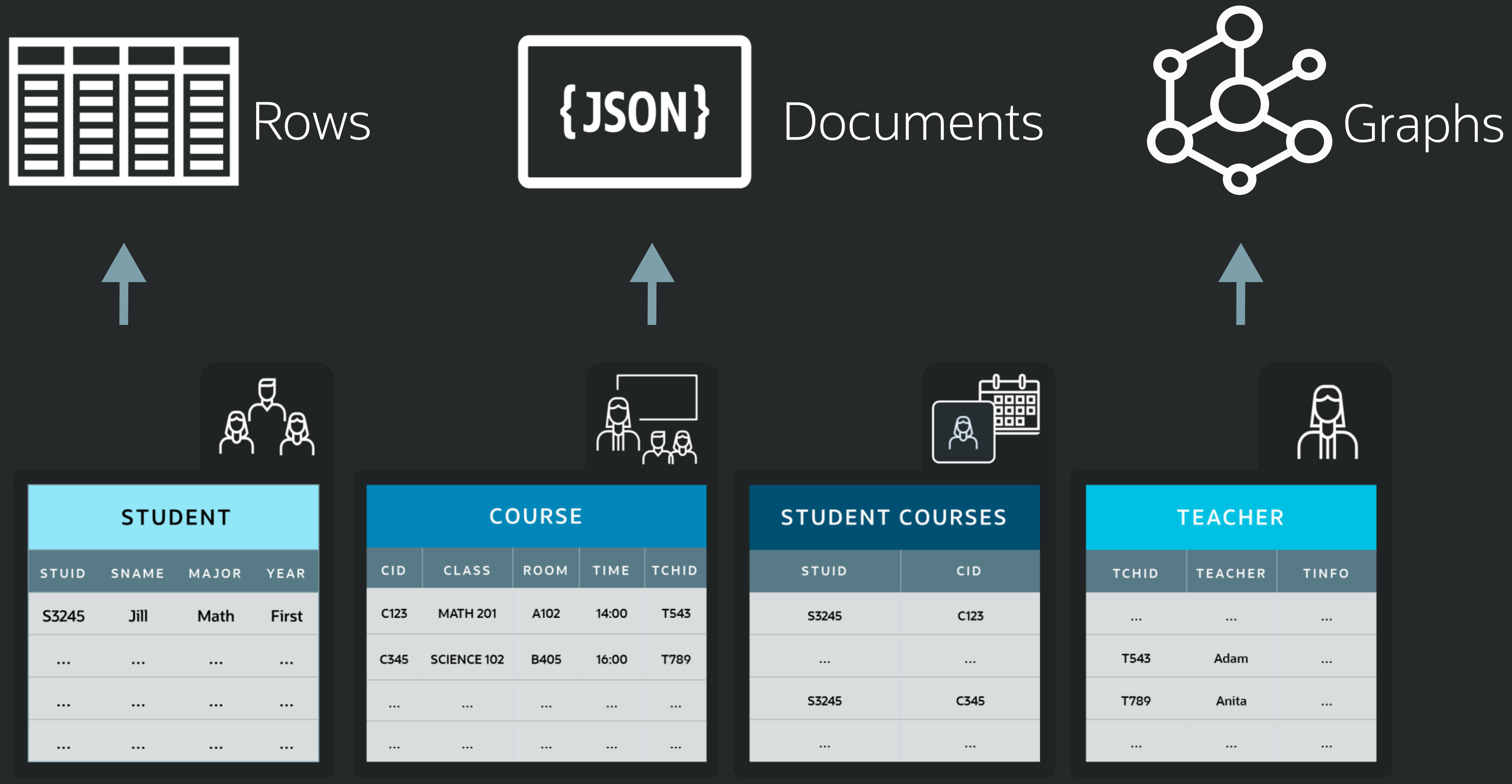
STUID	CID
S3245	C123
...	...
S3245	C345
...	...

TEACHER

TCHID	TEACHER	TINFO
...	...	...
T543	Adam	...
T789	Anita	...
...	...	...

Relational storage format

Generated usage formats



Rows

Documents

Graphs

STUDENT

STUID	SNAME	MAJOR	YEAR
S3245	Jill	Math	First
...	...	...	...
...	...	...	...
...	...	...	...

COURSE

CID	CLASS	ROOM	TIME	TCHID
C123	MATH 201	A102	14:00	T543
C345	SCIENCE 102	B405	16:00	T789
...	...	...	...	...
...	...	...	...	...

STUDENT COURSES

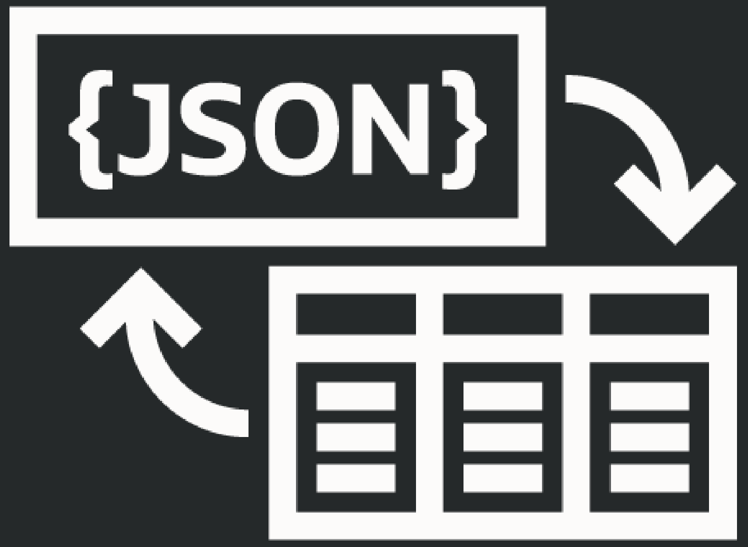
STUID	CID
S3245	C123
...	...
S3245	C345
...	...

TEACHER

TCHID	TEACHER	TINFO
...	...	...
T543	Adam	...
T789	Anita	...
...	...	...

Relational storage format

we need to define the usage intent



JSON Relational Duality Views

STUDENT

STUID	SNAME	MAJOR	YEAR
S3245	Jill	Math	First
...	...	...	...
...	...	...	...
...	...	...	...

COURSE

CID	CLASS	ROOM	TIME	TCHID
C123	MATH 201	A102	14:00	T543
C345	SCIENCE 102	B405	16:00	T789
...	...	...	...	...
...	...	...	...	...

STUDENT COURSES

STUID	CID
S3245	C123
...	...
S3245	C345
...	...

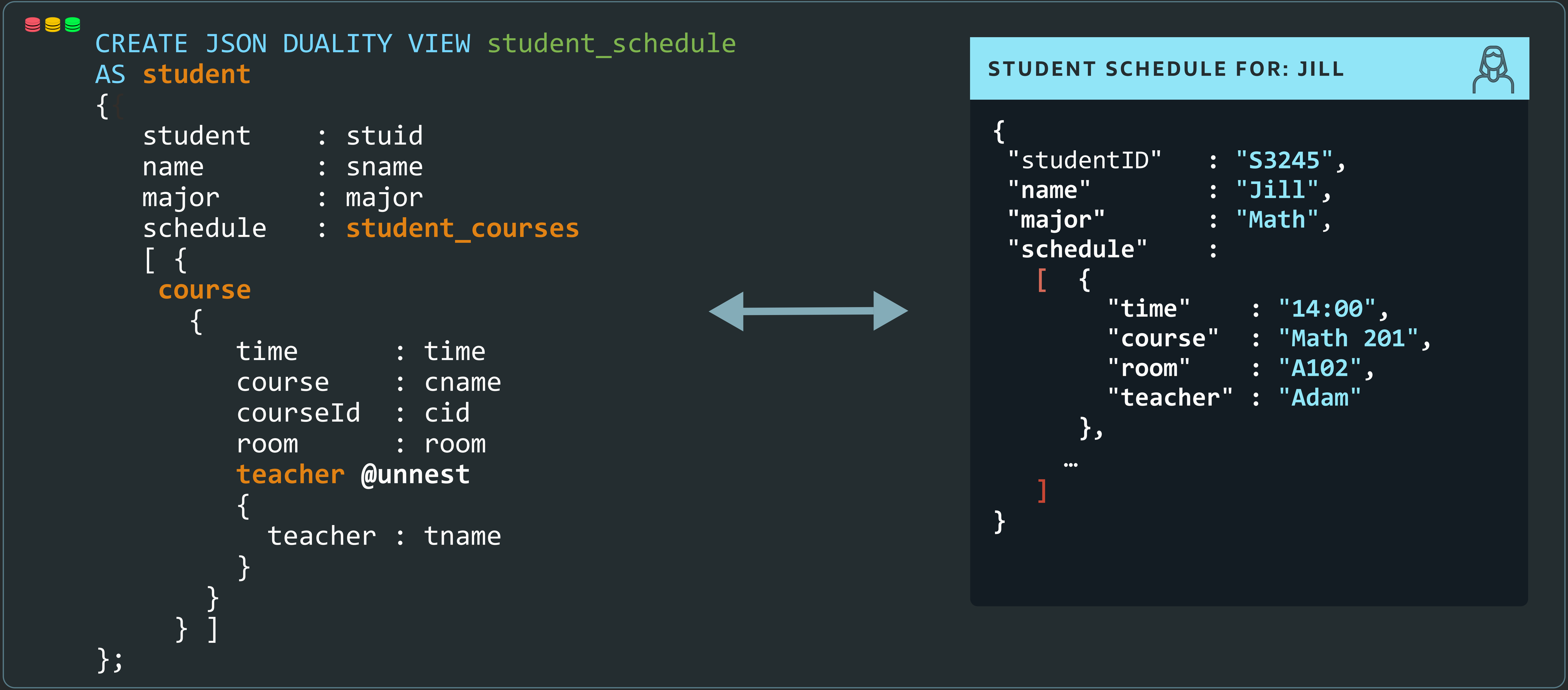
TEACHER

TCHID	TEACHER	TINFO
...	...	...
T543	Adam	...
T789	Anita	...
...	...	...

SCHEDULE FOR: JILL

```
{
 "studentID"    : "S3245",
 "name"         : "Jill",
 "major"        : "Math",
 "schedule"     :
   [  {
        "time"    : "14:00",
        "course"  : "Math 201",
        "room"    : "A102",
        "teacher" : "Adam"
      },
      {
        "time"    : "16:00",
        "course"  : "Science 102",
        "room"    : "B405",
        "teacher" : "Anita"
      }
   ]
}
```

easy definition

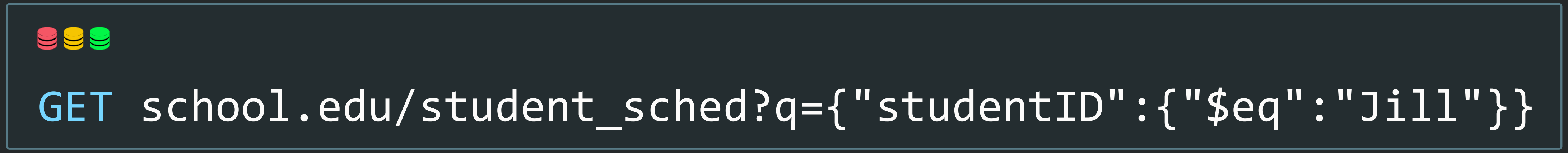


```
CREATE JSON DUALITY VIEW student_schedule
AS student
{
   student     : stuid
   name        : sname
   major       : major
   schedule    : student_courses
   [ {
    course
      {
         time      : time
         course    : cname
         courseId  : cid
         room      : room
         teacher @unnest
         {
           teacher : tname
         }
      }
    } ]
};
```

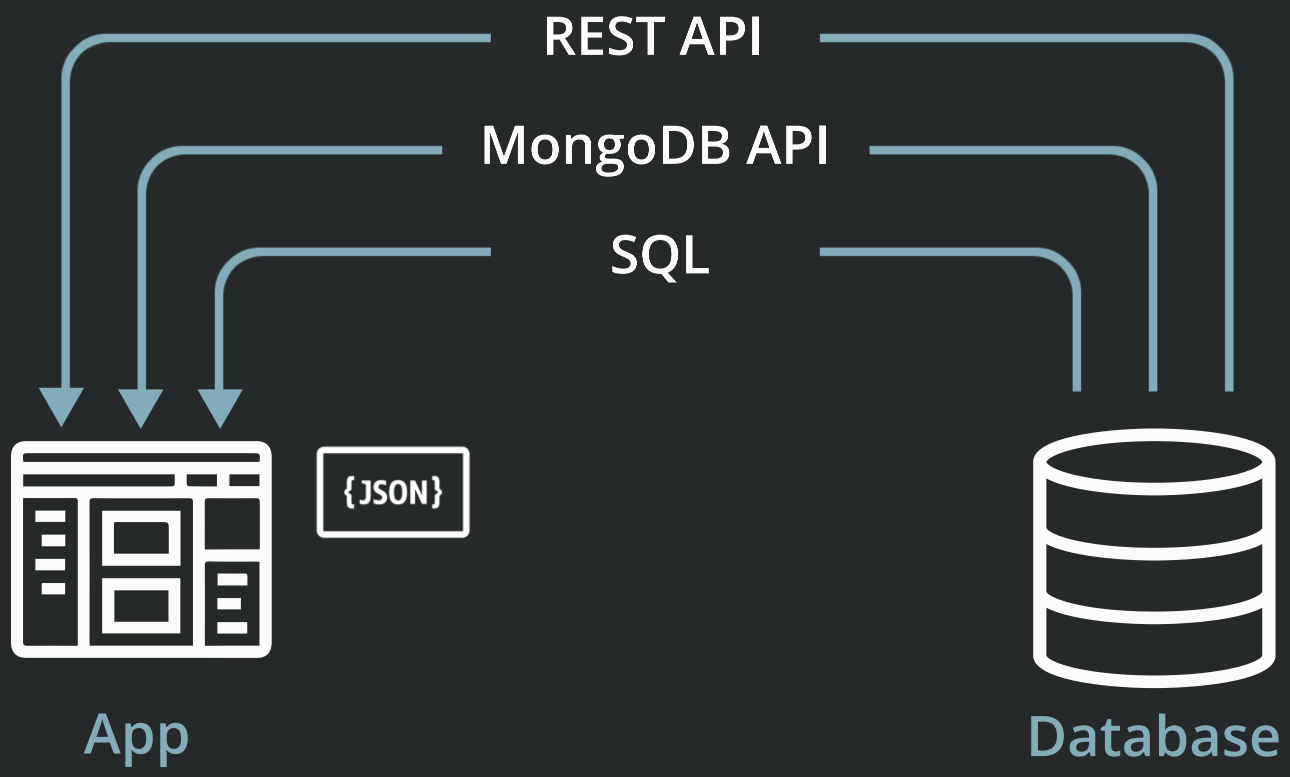
STUDENT SCHEDULE FOR: JILL

```
{
 "studentID"    : "S3245",
 "name"         : "Jill",
 "major"        : "Math",
 "schedule"     :
   [  {
         "time"    : "14:00",
         "course"  : "Math 201",
         "room"    : "A102",
         "teacher" : "Adam"
      },
     …
   ]
}
```

document APIs for developers



```
GET school.edu/student_sched?q={"studentID":{"$eq":"Jill"}}
```



REST API

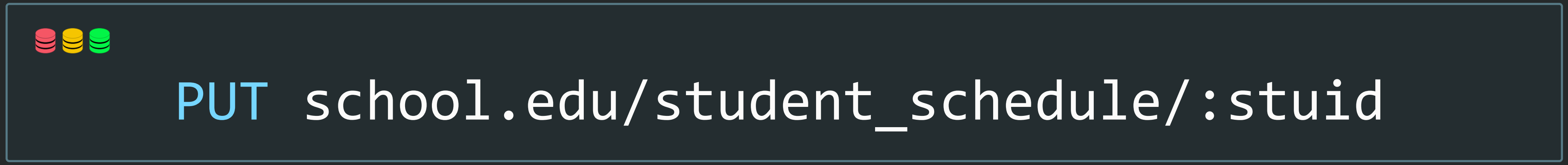
MongoDB API

SQL

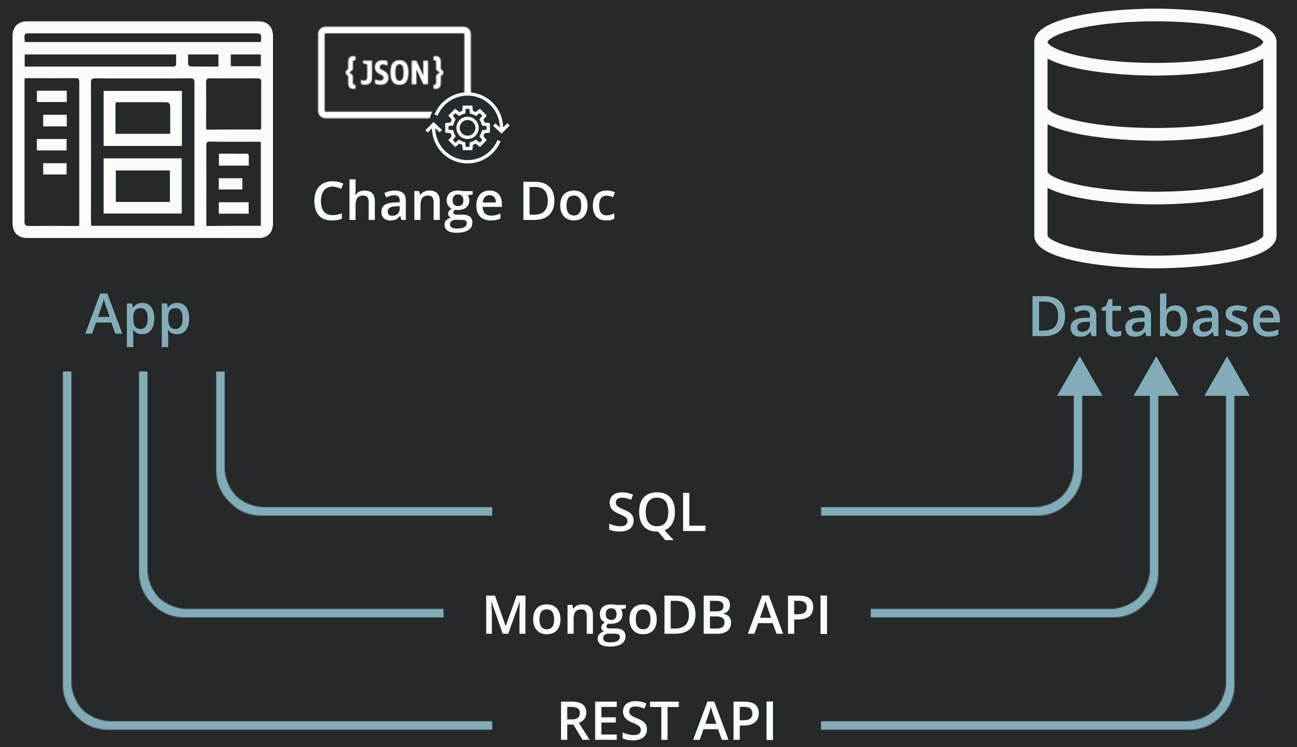
{JSON}

App

Database



```
PUT school.edu/student_schedule/:stuid
```



it is still relational data

STUDENT SCHEDULE FOR: JILL

```
{
 "studentID"    : "S3245",
 "name"         : "Jill",
 "major"        : "Math",
 "schedule"     :
   [  {
       "time"    : "14:00",
       "course"  : "Math 201",
       "room"    : "A102",
       "teacher" : "Adam"
      },
      {
       "time"    : "16:00",
       "course"  : "Science 102",
       "room"    : "B405",
       "teacher" : "Anita"
      }
   ]
}
```

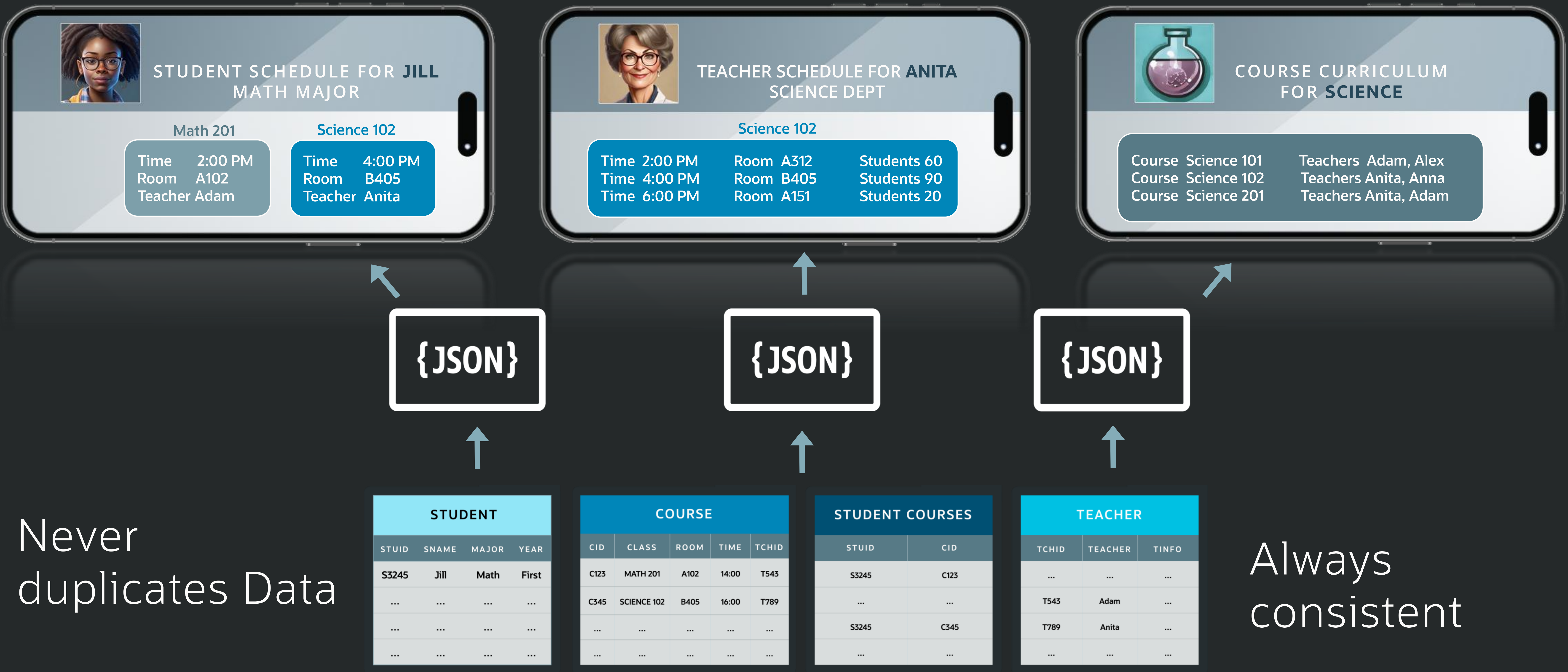
Looks duplicated but is not

STUDENT SCHEDULE FOR: LUCAS

```
{
 "studentID"    : "S4356",
 "name"         : "Lucas",
 "major"        : "Engineering",
 "schedule"     :
   [  {
       "time"    : "14:00",
       "course"  : "Math 201",
       "room"    : "A102",
       "teacher" : "Adam"
      },
      {
       "time"    : "18:00",
       "course"  : "Physics",
       "room"    : "A115",
       "teacher" : "Alex"
      }
   ]
}
```

different JSON for different use cases

data never duplicated



STUDENT SCHEDULE FOR **JILL**
MATH MAJOR

Math 201
Time 2:00 PM
Room A102
Teacher Adam

Science 102
Time 4:00 PM
Room B405
Teacher Anita

TEACHER SCHEDULE FOR **ANITA**
SCIENCE DEPT

Science 102
Time 2:00 PM Room A312 Students 60
Time 4:00 PM Room B405 Students 90
Time 6:00 PM Room A151 Students 20

COURSE CURRICULUM
FOR **SCIENCE**

Course Science 101 Teachers Adam, Alex
Course Science 102 Teachers Anita, Anna
Course Science 201 Teachers Anita, Adam

STUDENT

STUID	SNAME	MAJOR	YEAR
S3245	Jill	Math	First
...	...	...	...
...	...	...	...
...	...	...	...

COURSE

CID	CLASS	ROOM	TIME	TCHID
C123	MATH 201	A102	14:00	T543
C345	SCIENCE 102	B405	16:00	T789
...	...	...	...	...
...	...	...	...	...

STUDENT COURSES

STUID	CID
S3245	C123
...	...
S3245	C345
...	...

TEACHER

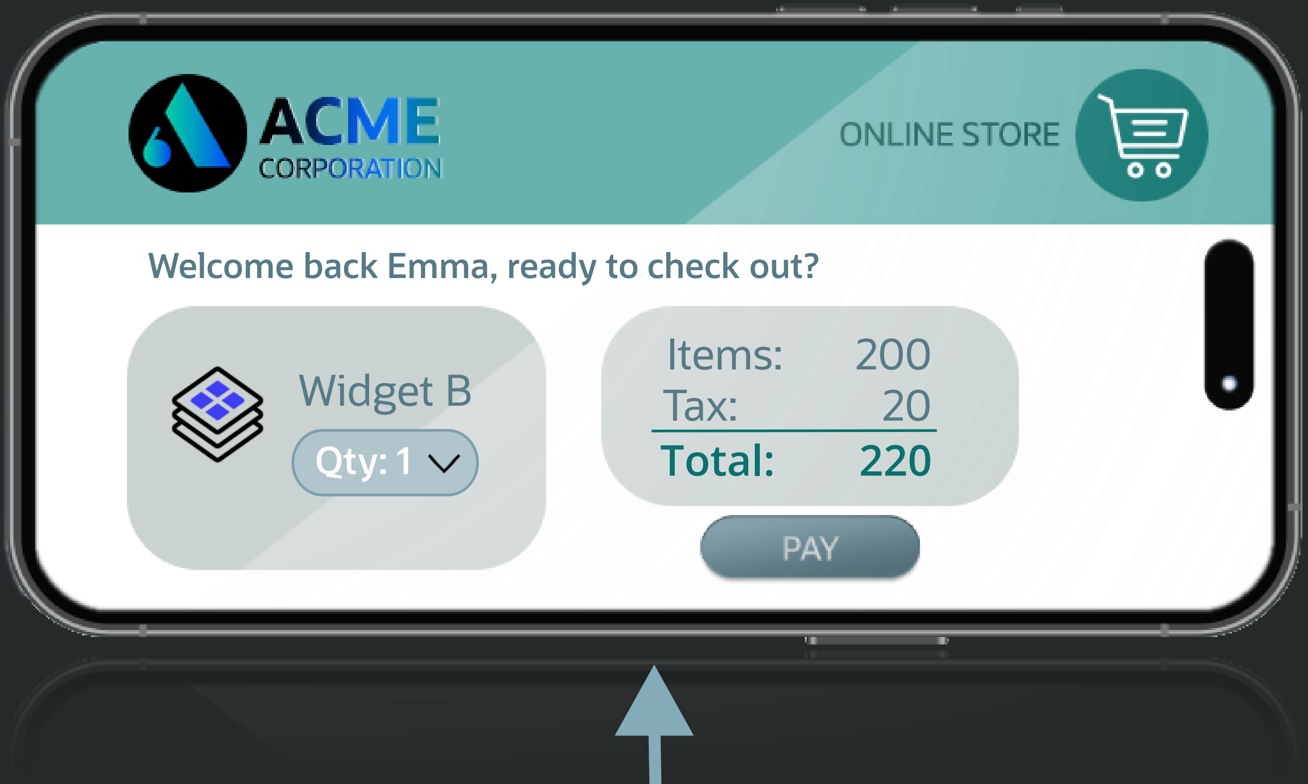
TCHID	TEACHER	TINFO
...	...	...
T543	Adam	...
T789	Anita	...
...	...	...

Never duplicates Data

Always consistent

modernise legacy applications

no database rework



Welcome back Emma, ready to check out?

Widget B

Qty: 1

Items:	200
Tax:	20
Total:	220

PAY

{JSON}

ORDERS

ORDER	NAME	PROD_ID	PRICE
345	EMMA	948	$200
...	...	...	...
.	...	...	...
...	...	...	...

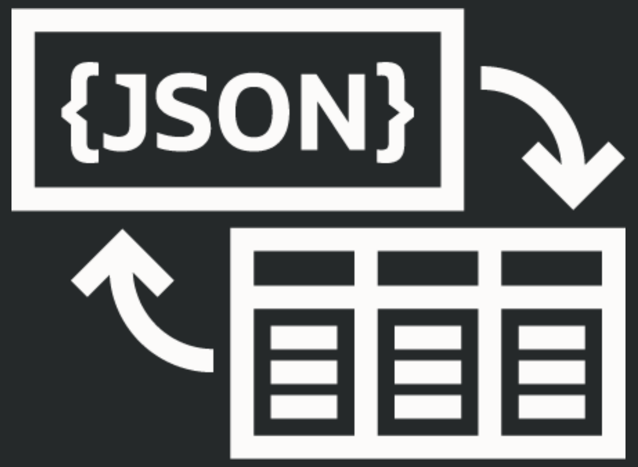
CUSTOMERS

NAME	ADDRS	PHONE
....	...	...
EMMA	...	(415)...
...	...	...
...	...	...

PRODUCTS

PROD_ID	PROD_NAME
...	...
...	...
948	WIDGET B
...	...

"Isn't this just an ORM?"



architectural simplicity

no extra layer

ACID / locking controls

any app, any language, no app tier code

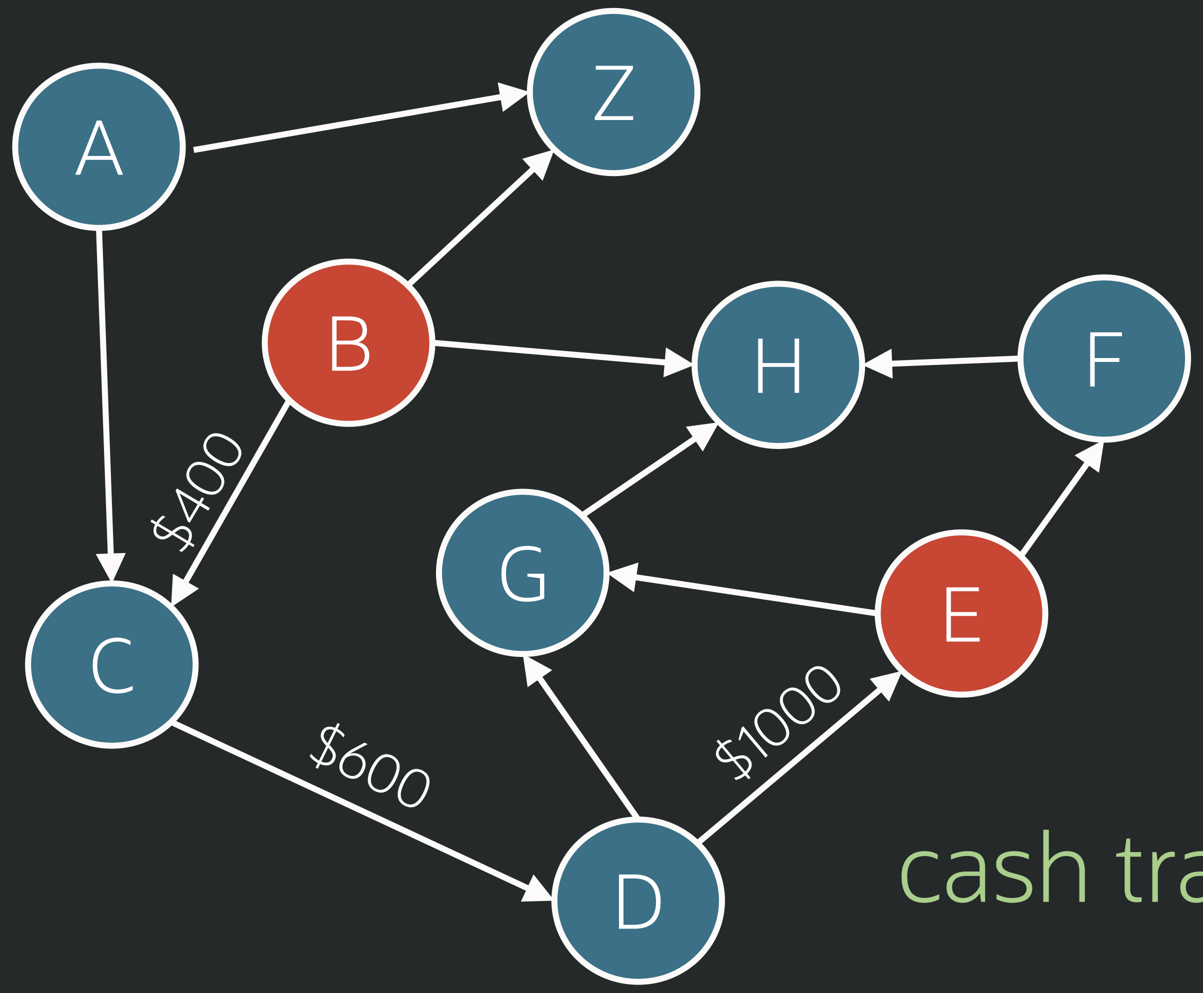
easy adoption

single round trip

bring the same principle to graph

relational storage with graph usage intent

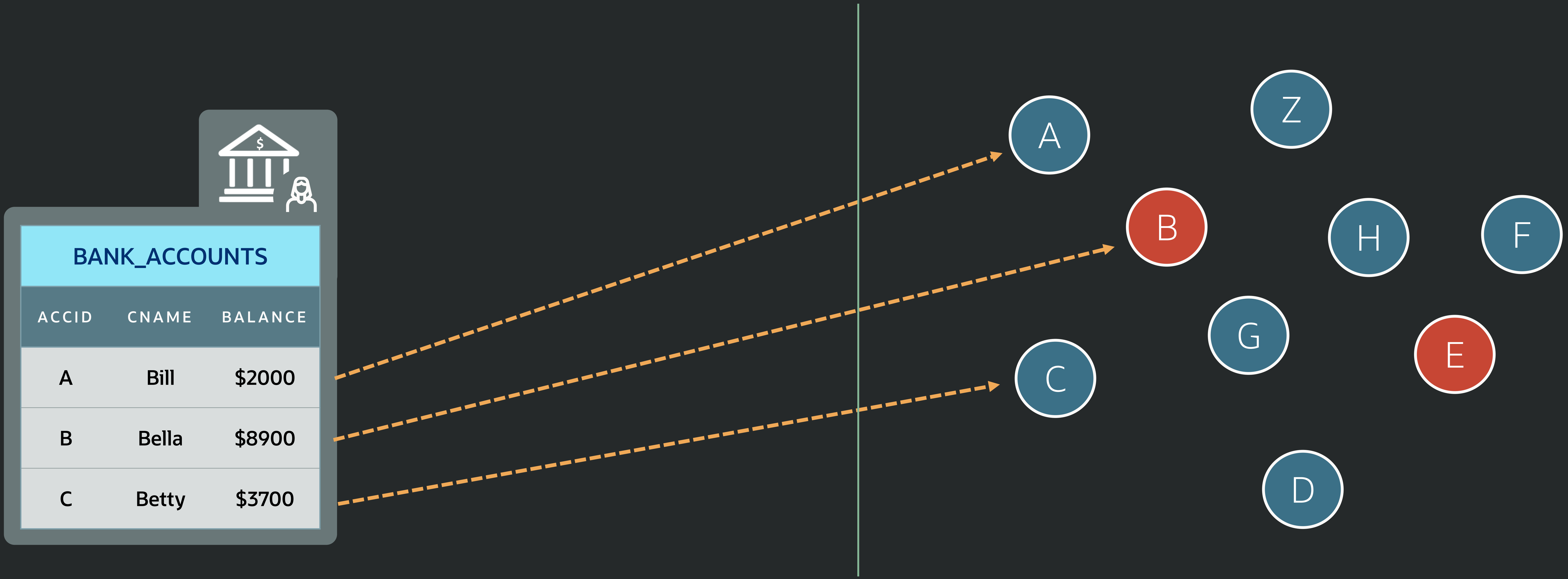
How to write this in SQL?



cash transfers from "B" to "E" ?

property graph view

bank accounts → vertices



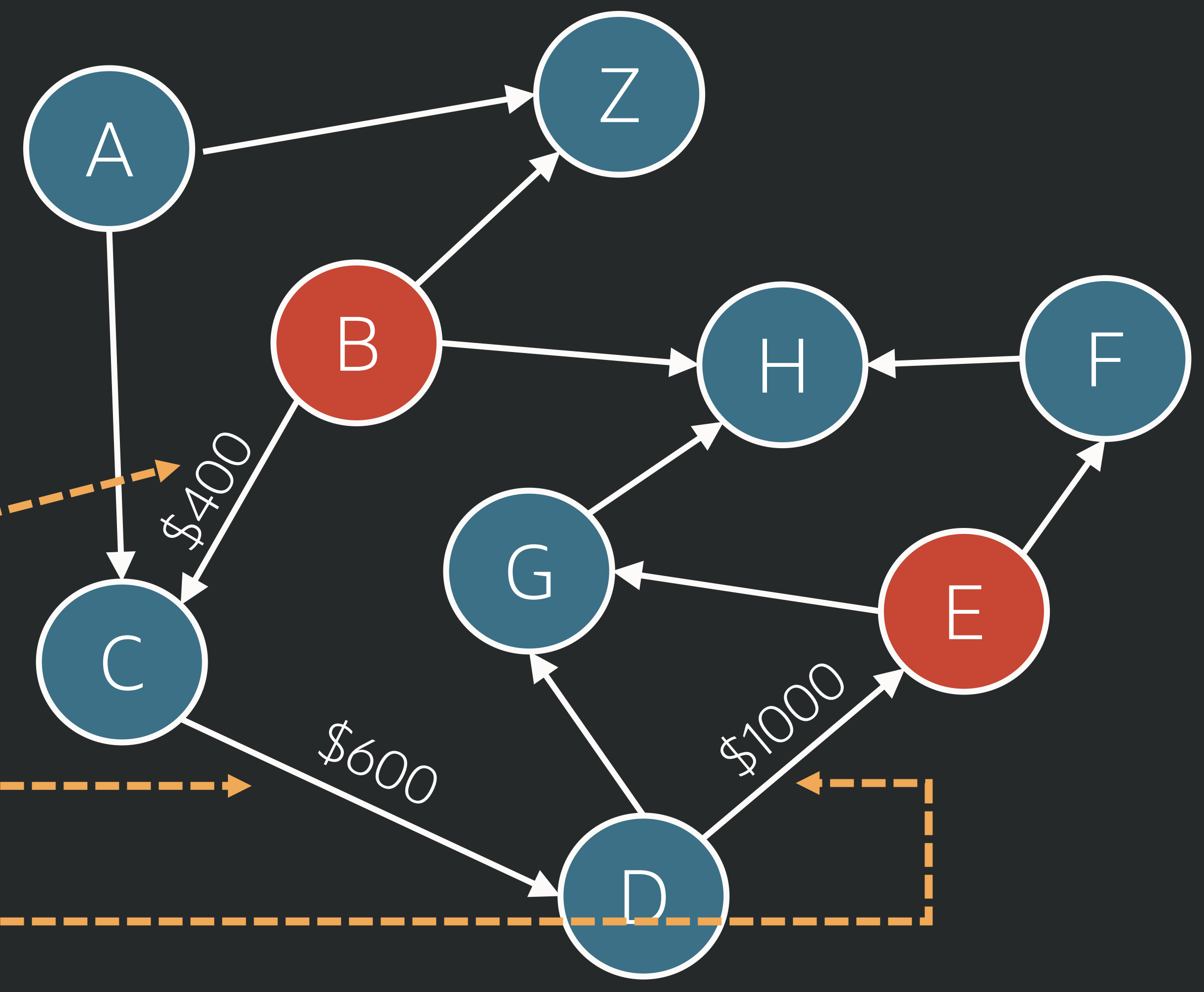
BANK_ACCOUNTS

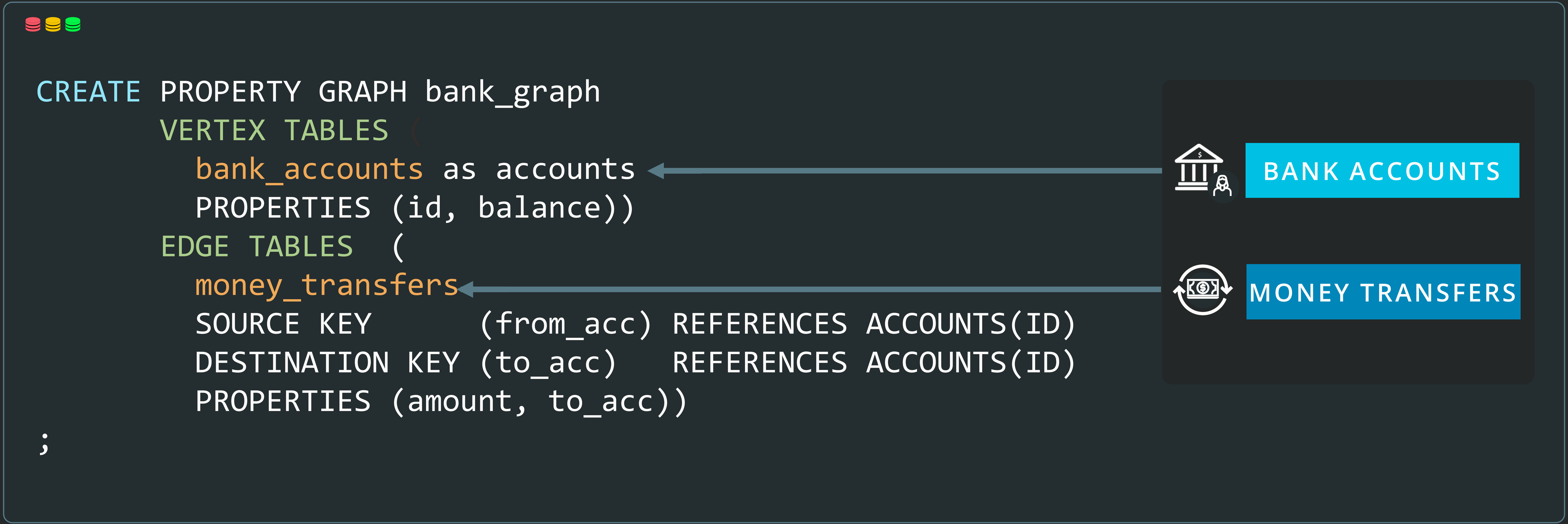
ACCID	CNAME	BALANCE
A	Bill	$2000
B	Bella	$8900
C	Betty	$3700

transfers → edges

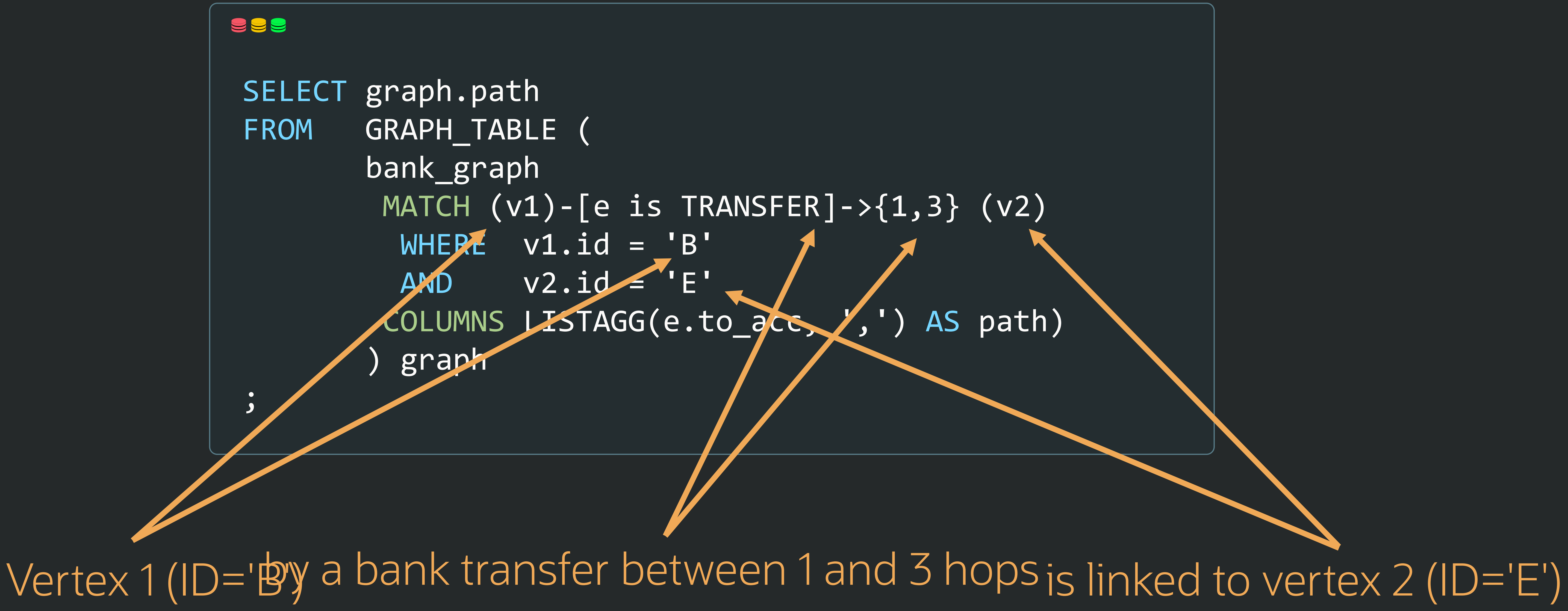
BANK_ACCOUNTS		
ACCID	CNAME	BALANCE
A	Bill	$2000
B	Bella	$8900
C	Betty	$3700

MONEY TRANSFERS		
FROM_ACC	TO_ACC	AMOUNT
B	C	$400
C	D	$600
D	E	$1000



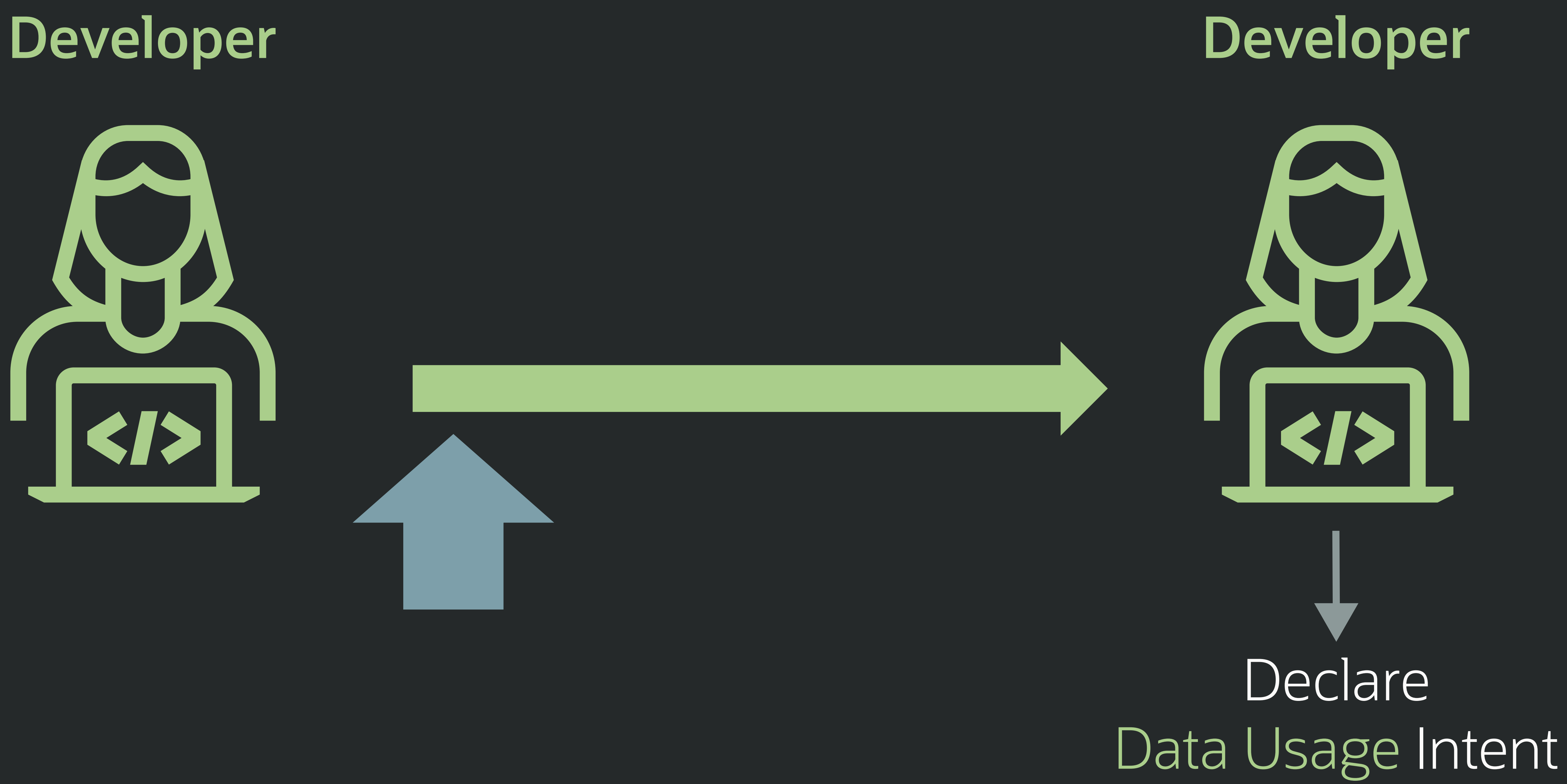


```
CREATE PROPERTY GRAPH bank_graph
       VERTEX TABLES (
         bank_accounts as accounts
         PROPERTIES (id, balance))
       EDGE TABLES  (
         money_transfers
         SOURCE KEY      (from_acc) REFERENCES ACCOUNTS(ID)
         DESTINATION KEY (to_acc)   REFERENCES ACCOUNTS(ID)
         PROPERTIES (amount, to_acc))
;
```



```
SELECT graph.path
FROM   GRAPH_TABLE (
       bank_graph
        MATCH (v1)-[e is TRANSFER]->{1,3} (v2)
         WHERE  v1.id = 'B'
         AND    v2.id = 'E'
        COLUMNS LISTAGG(e.to_acc, ',') AS path)
       ) graph
;
```

Vertex 1 (ID='B') by a bank transfer between 1 and 3 hops is linked to vertex 2 (ID='E')



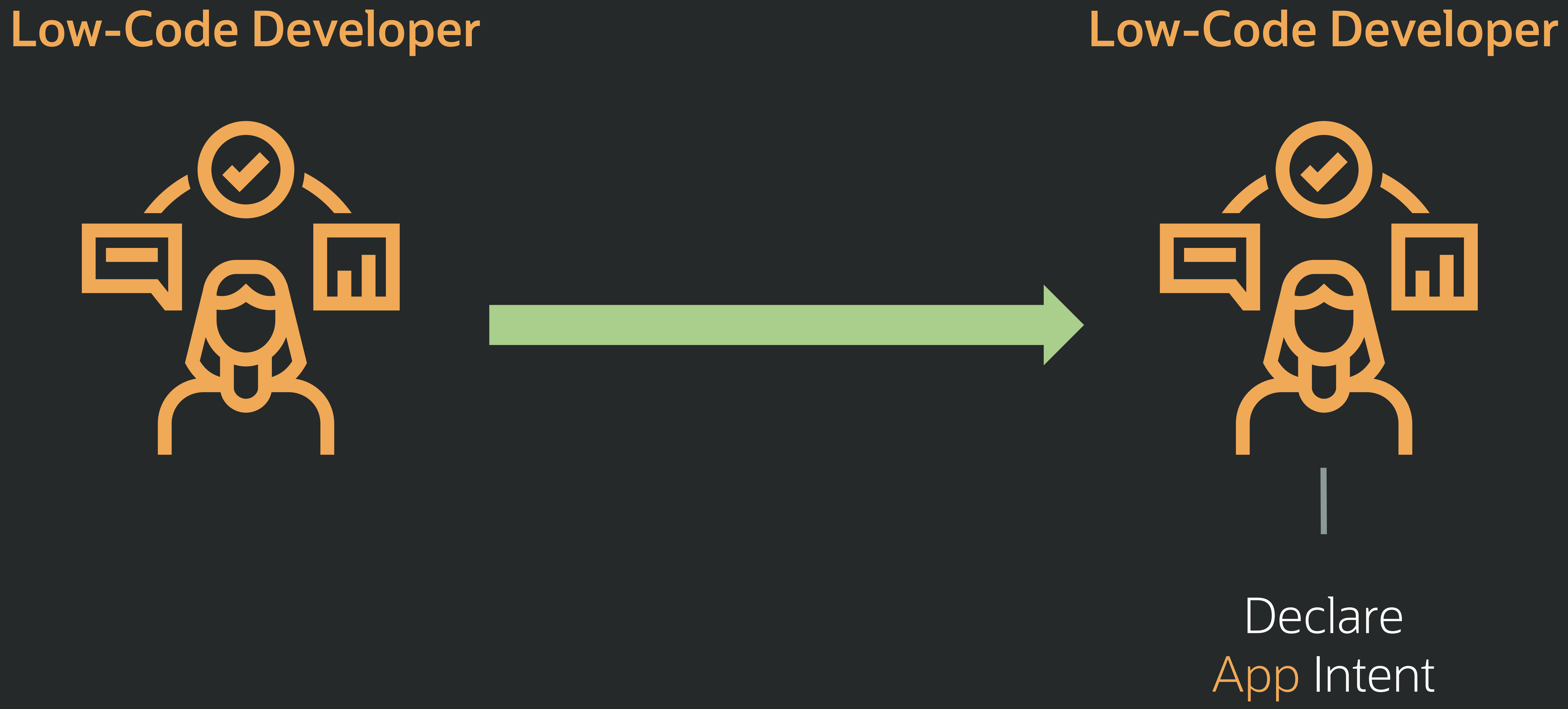
Developer

Developer

Declare
Data Usage Intent

1) generate data for usage ✓

2) generate apps not code them



Low-Code Developer

Low-Code Developer

Declare
App Intent

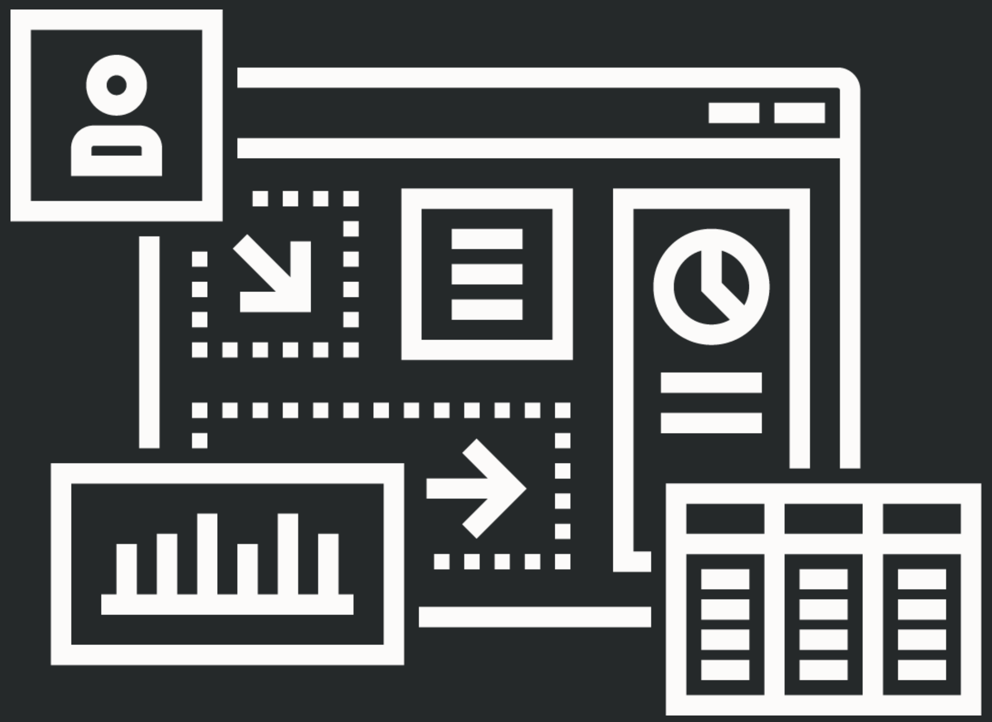
low code is more productive but …

… most historical attempts have failed

for 2 reasons

1) generates the code, maintain the code

2) the leap to enterprise class



Over 2 million APEX apps
3K new apps are every day

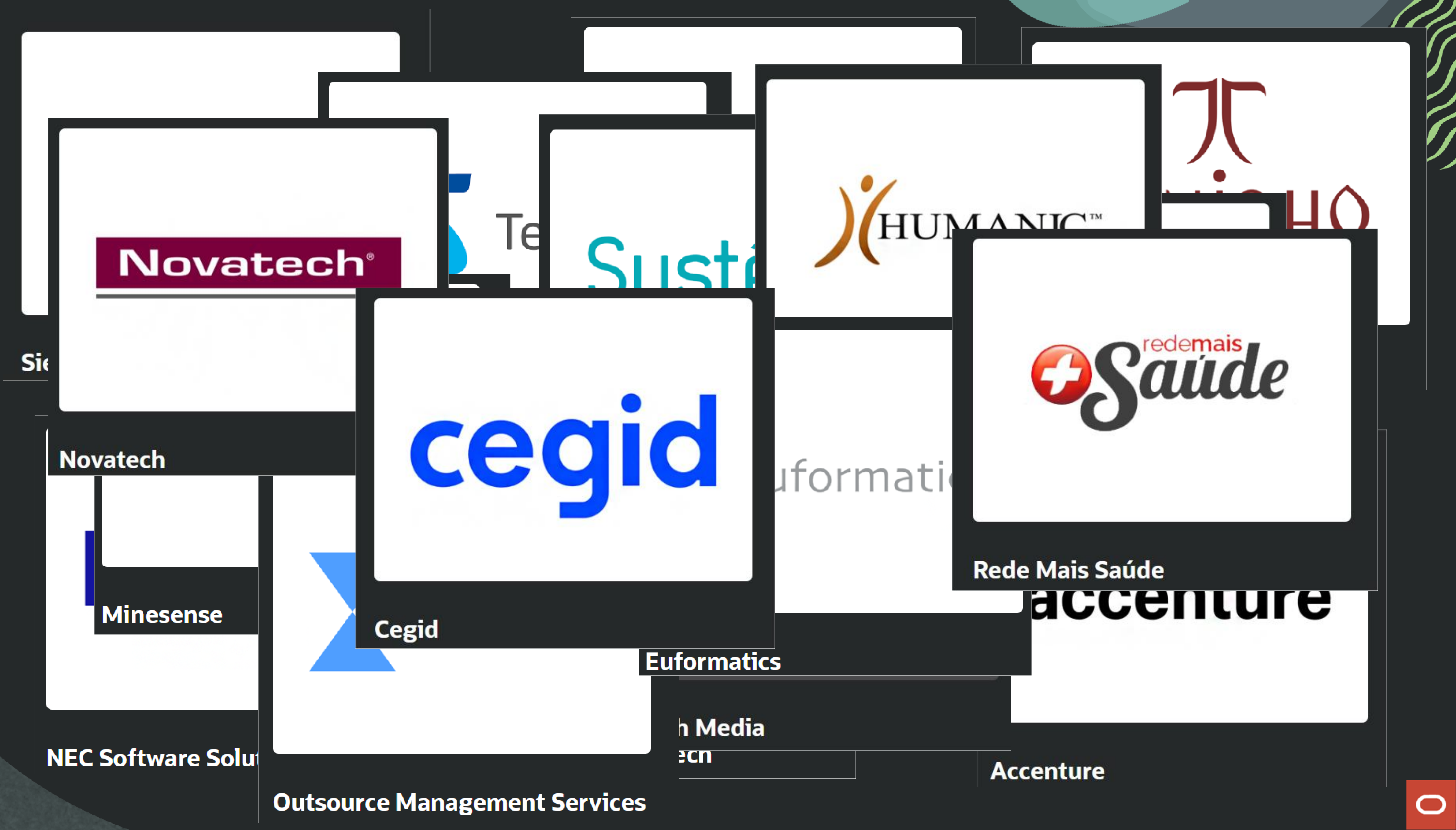
APEX solves both issues

metadata

^ generates the code, maintain the code

metadata

enterprise grade apps



Sie

Novatech

Minesense

NEC Software Solu

Cegid

Outsource Management Services

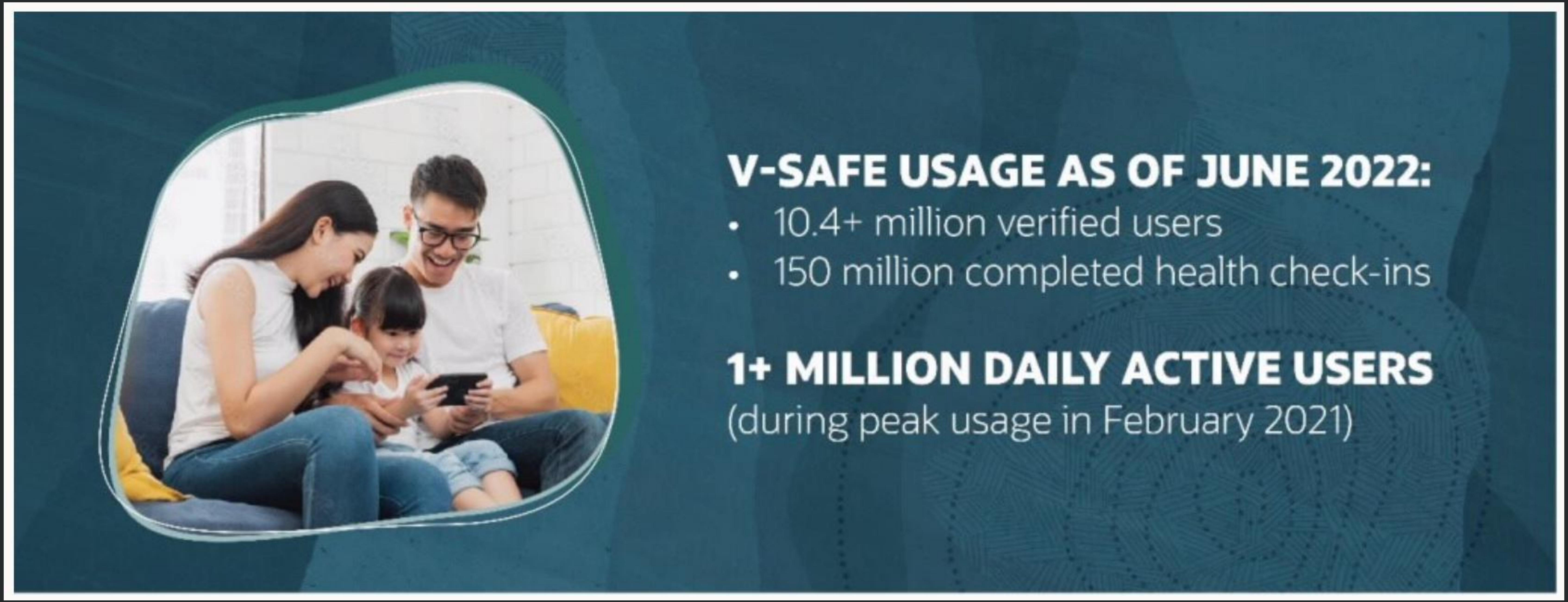
Euformatics

n Media

ech

Rede Mais Saúde

Accenture



V-SAFE USAGE AS OF JUNE 2022:

- 10.4+ million verified users
- 150 million completed health check-ins

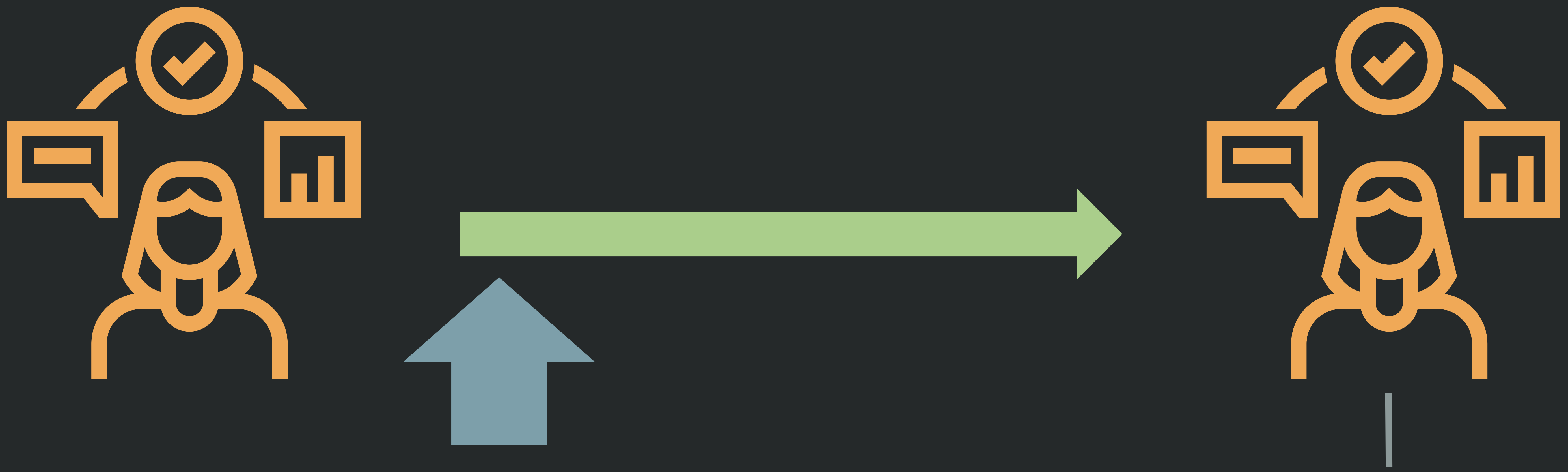
1+ MILLION DAILY ACTIVE USERS

(during peak usage in February 2021)

low code is not just APEX

low code across all domains

Low-code data modeling	Semantic Modeling	Data Modeling	ML Modeling	Graph Modeling
Low-code data motion	Data Sharing	Data Integration	Data Catalog	REST Services
Low-code data analysis	Data Analysis	SQL Worksheet	Notebooks	JSON Worksheet



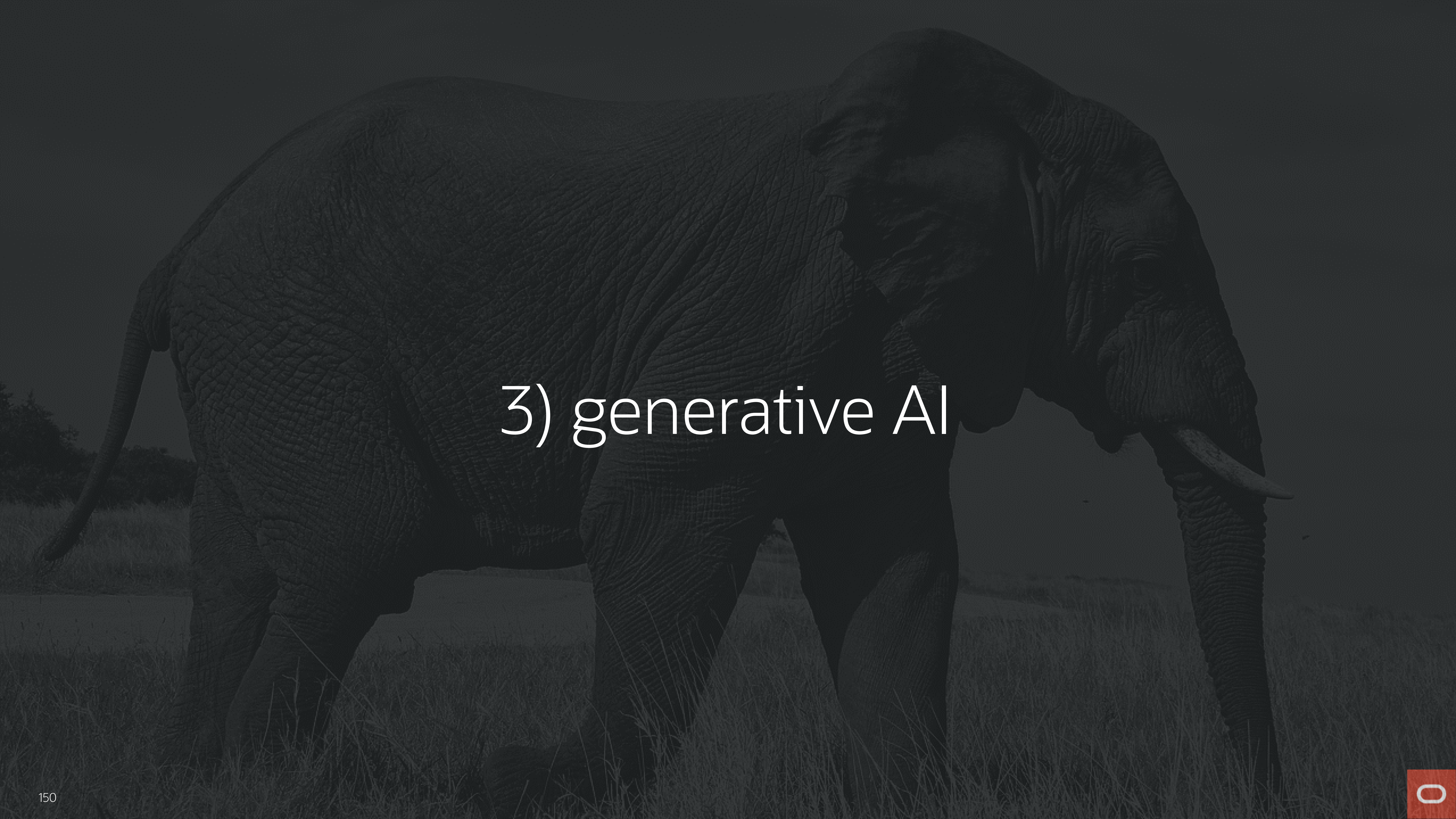
Low-Code Developer

Low-Code Developer

Declare
App Intent

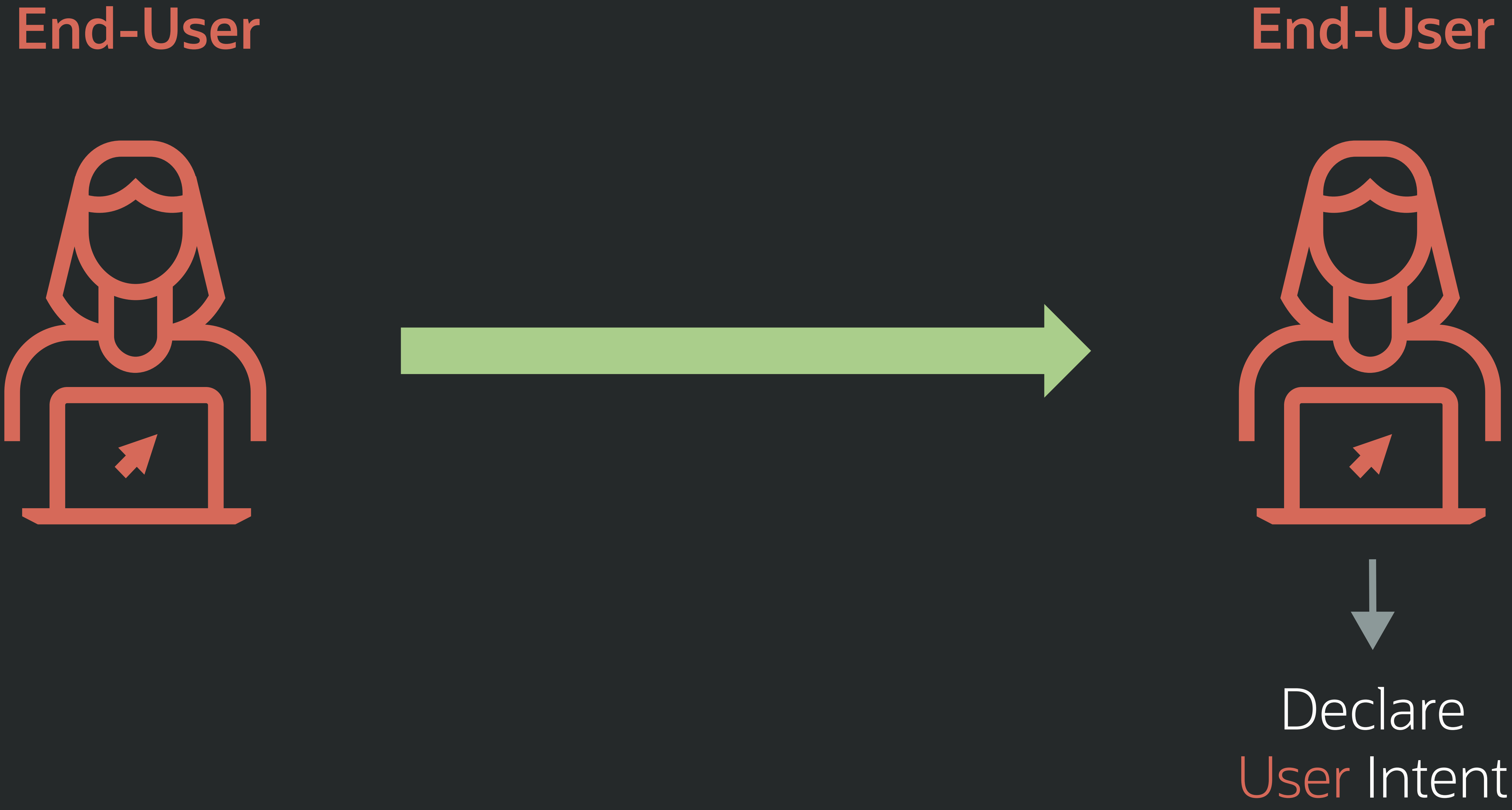
1) generate data for usage ✓

2) generate apps not code them ✓



3) generative AI

new wave of declarative intent



End-User

End-User

Declare

User Intent

exploit any data with natural language

exploit any data …

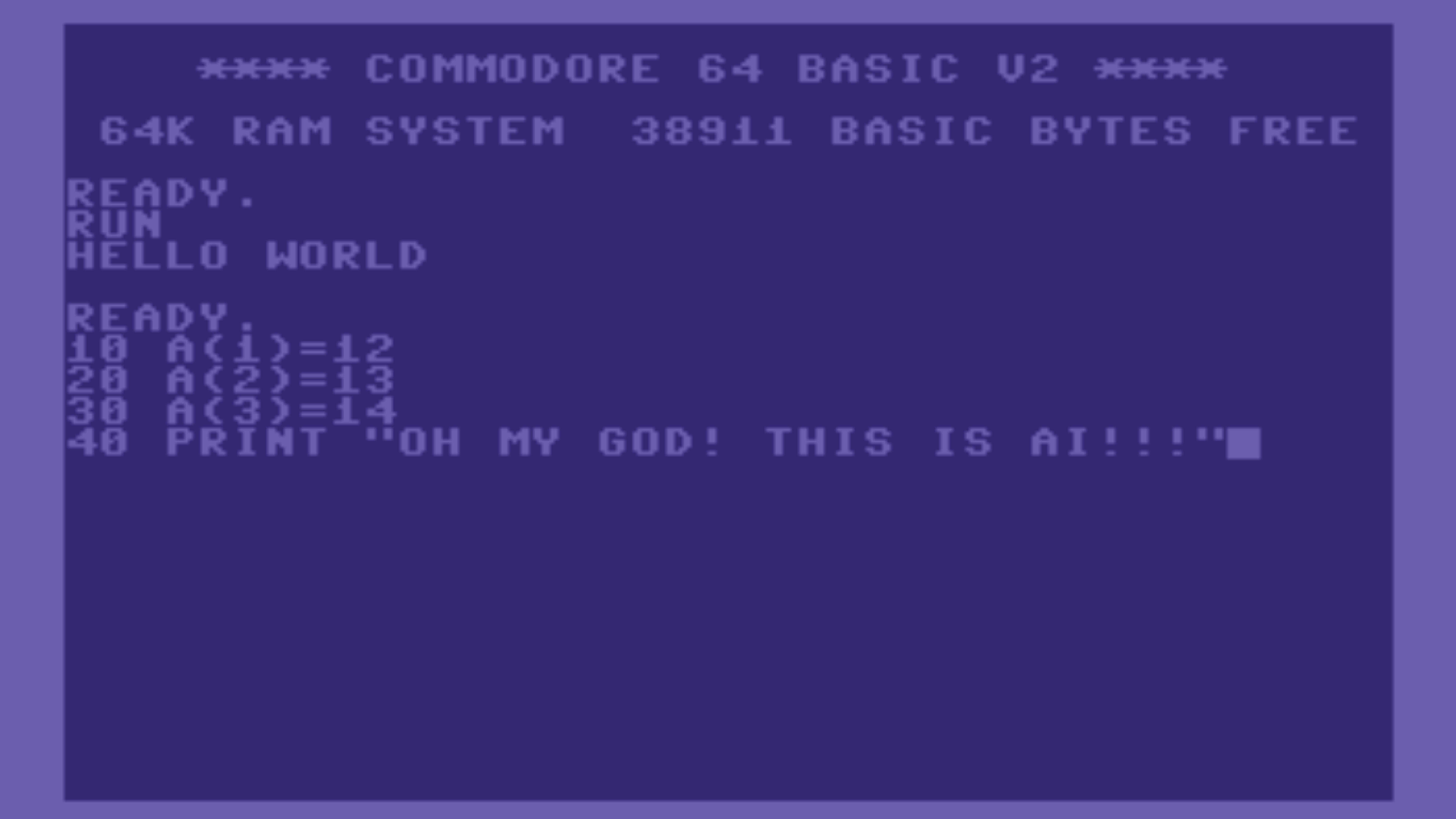
the basics

"new" data type

50 21 16 42 33

AI Vectors

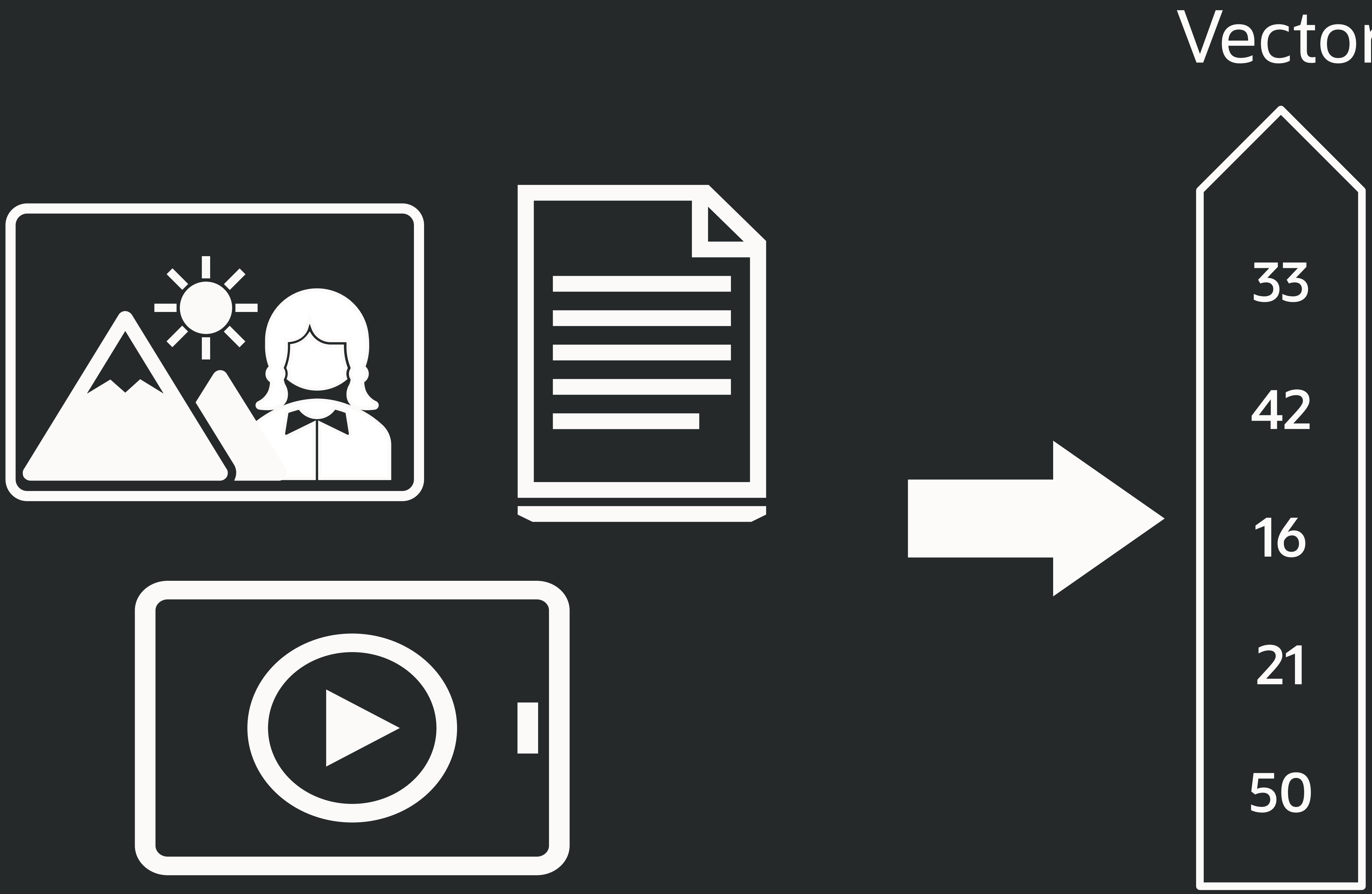
"Dude...you invented arrays."



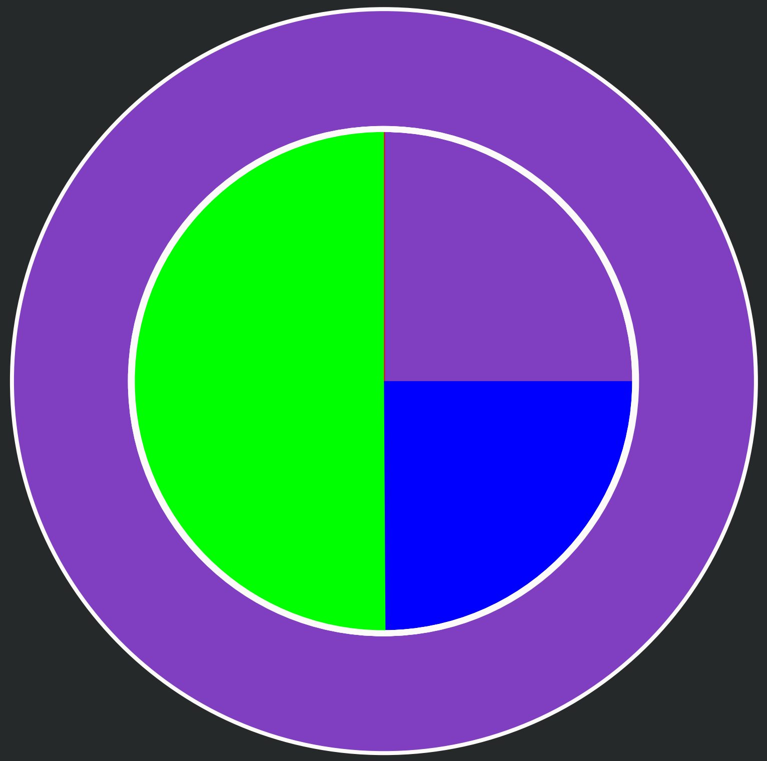
```
    **** COMMODORE 64 BASIC V2 ****

 64K RAM SYSTEM  38911 BASIC BYTES FREE

READY.
RUN
HELLO WORLD

READY.
10 A(1)=12
20 A(2)=13
30 A(3)=14
40 PRINT "OH MY GOD! THIS IS AI!!!"
```

key point

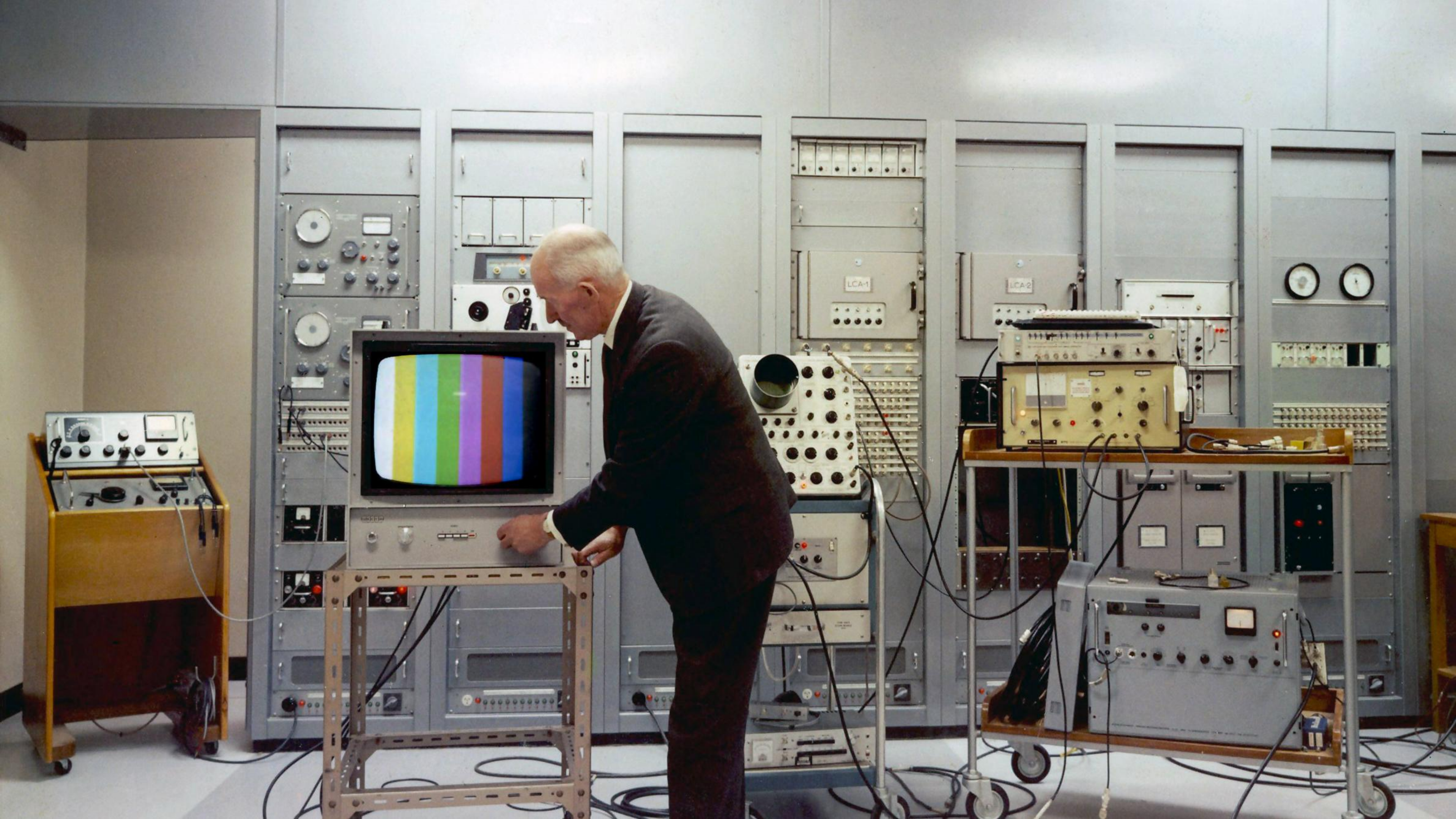
map data to a vector


Vector
33
42
16
21
50

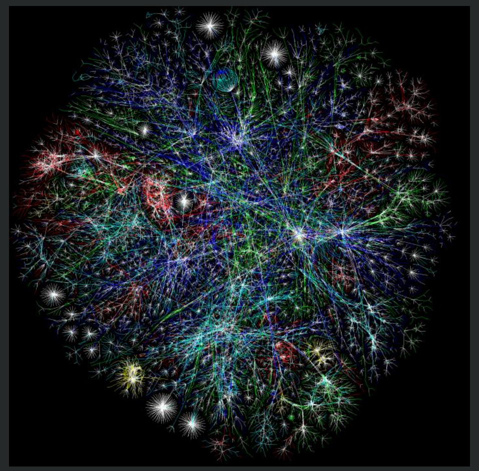
this is not new



"purple" = {r,g,b} = {25%, 50%, 75%}

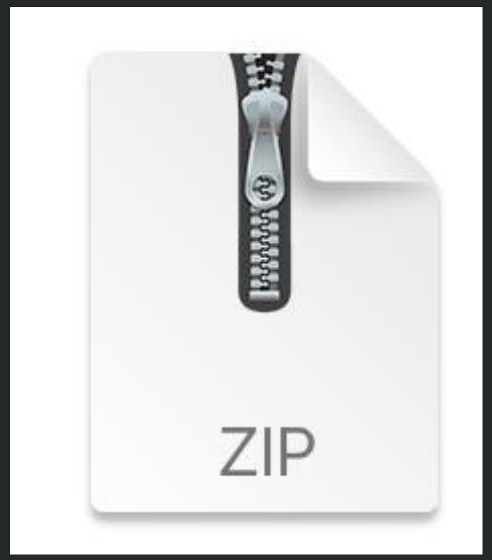

LCA-1
LCA-2

so what changed?



10TB raw data

10,000 GPUs
weeks of compute
$$$$$



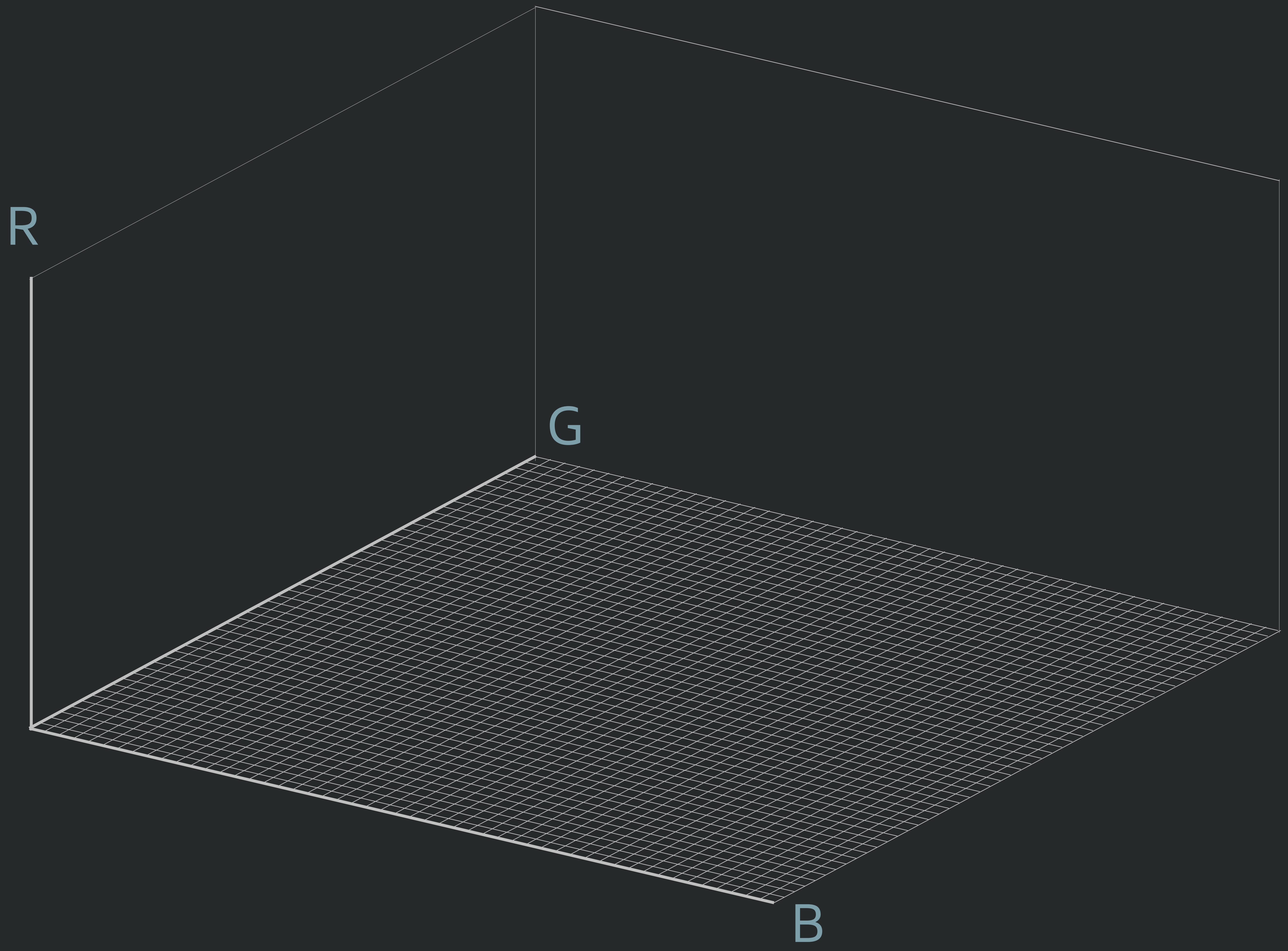
100GB
vectors

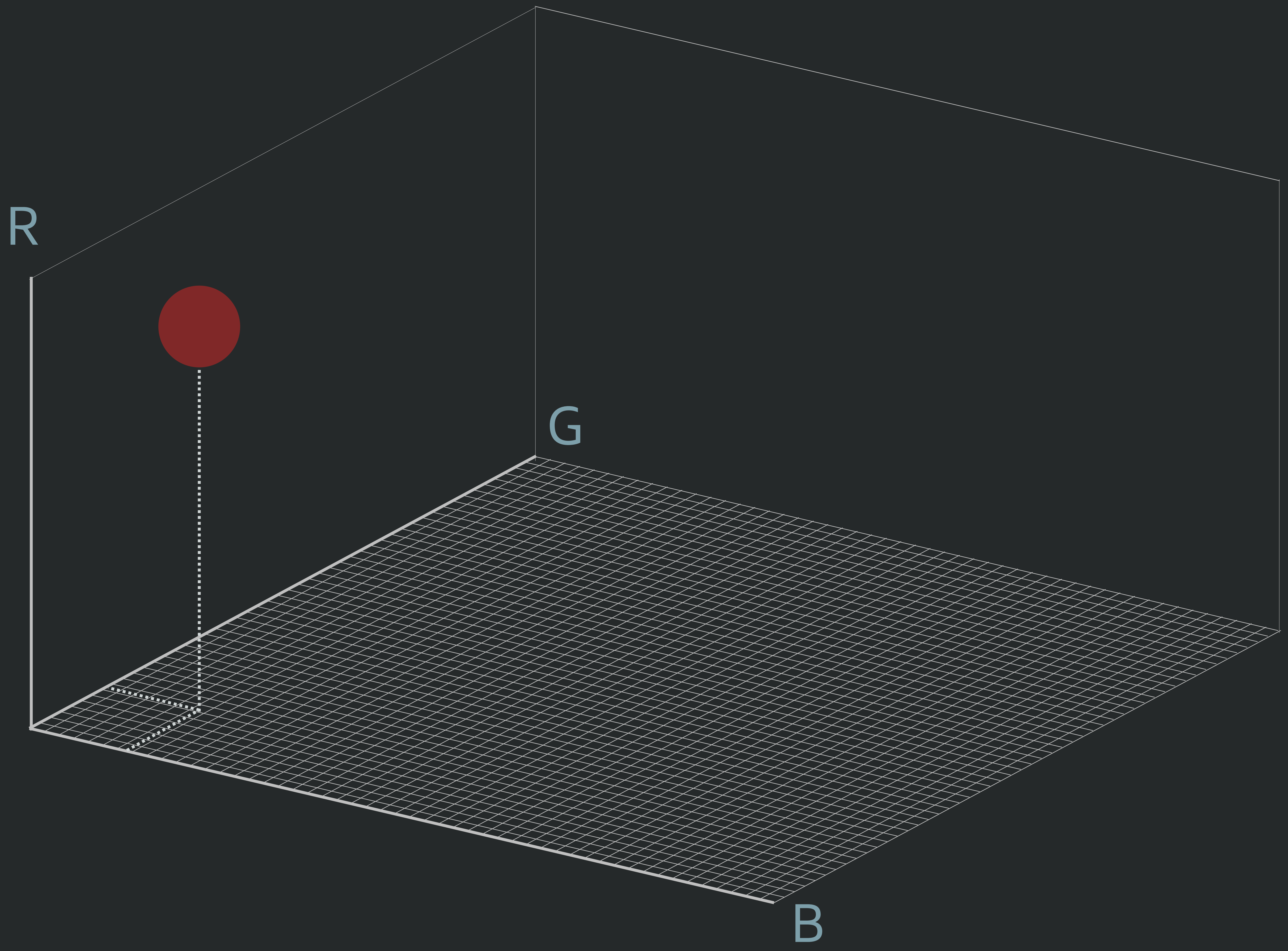
Credit: Intro to Large Language Models, Andrej Karpathy

the model is the magic

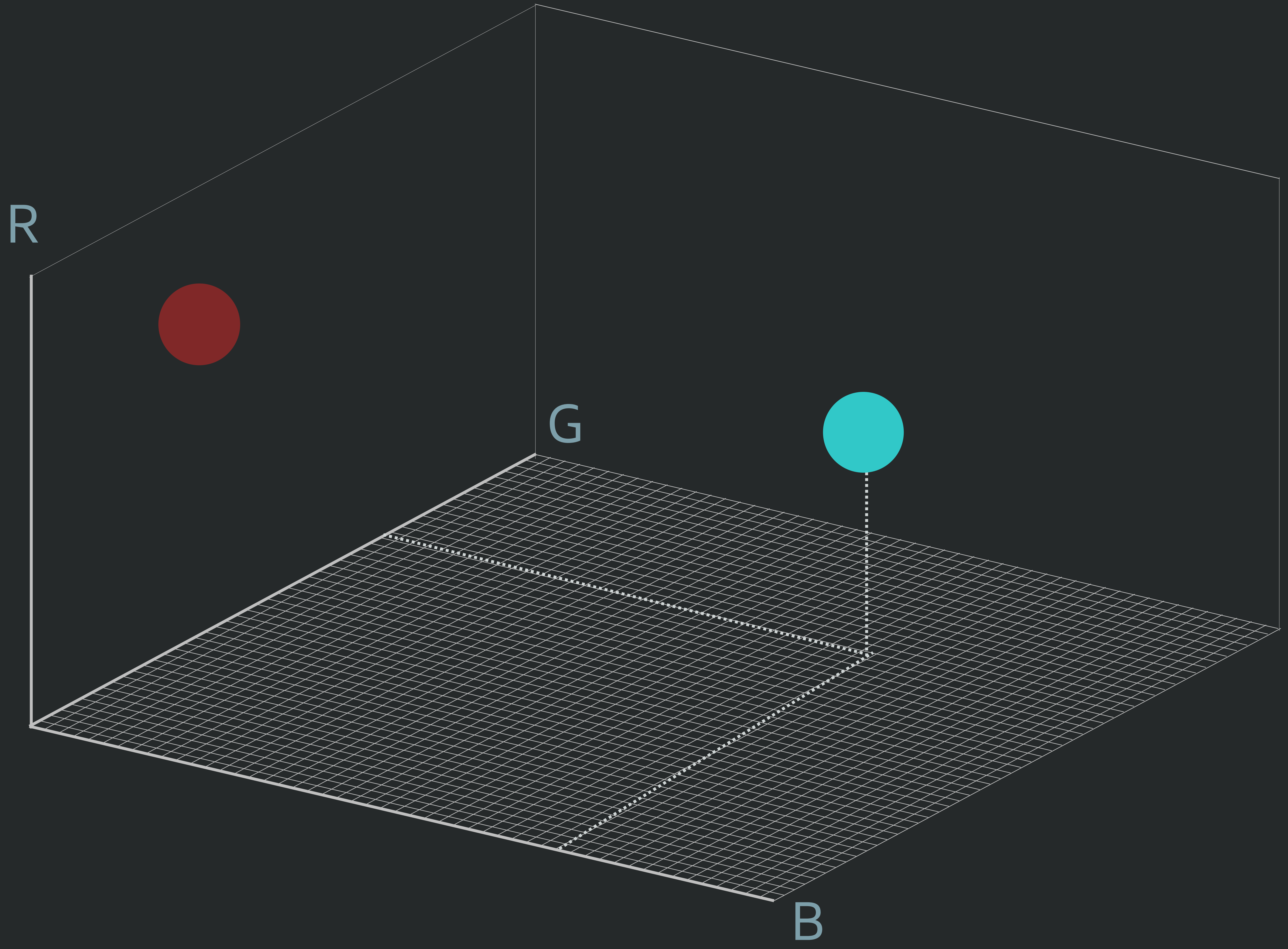
why ?

back to our RGB


R
G
B

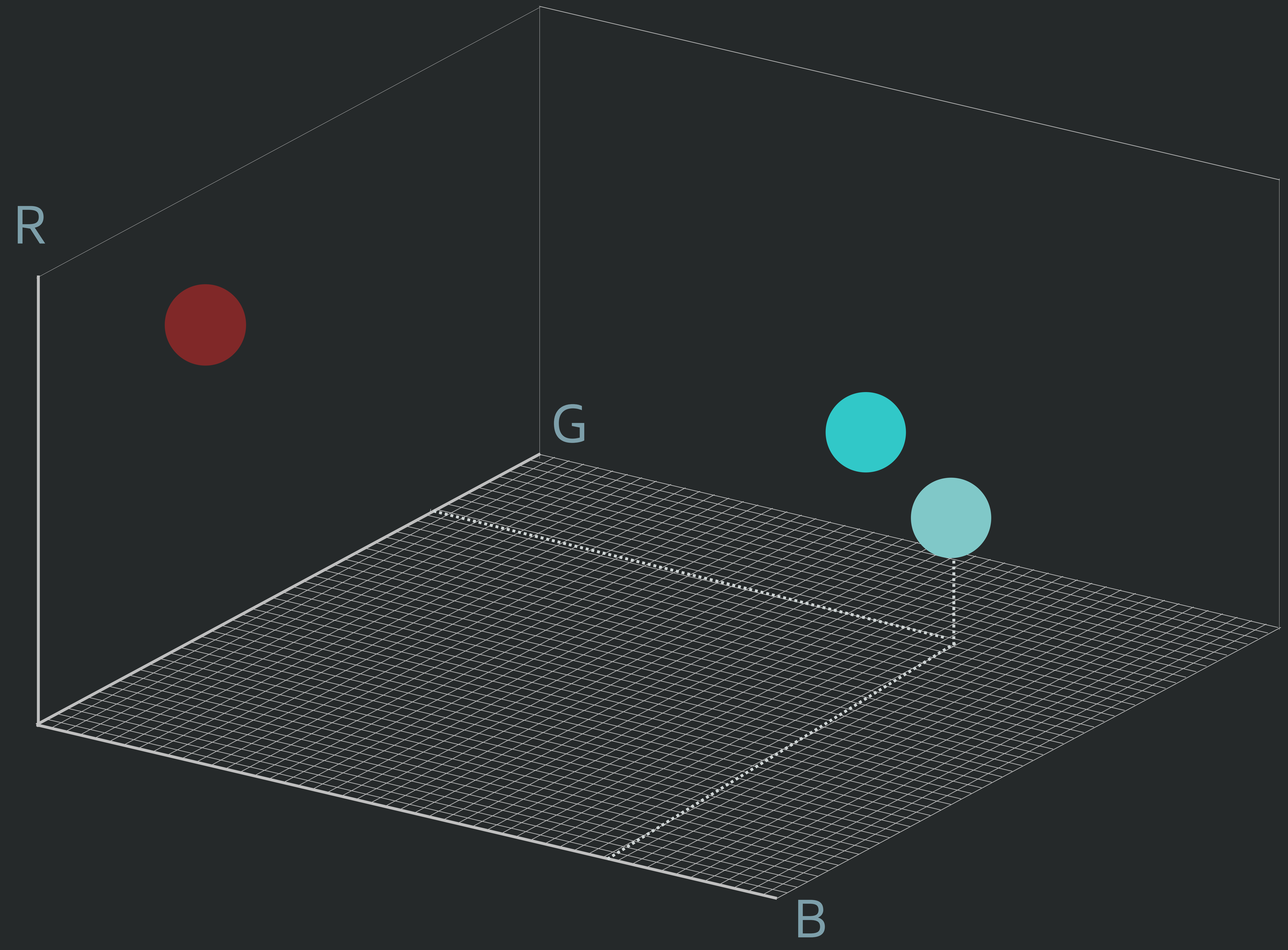


● {75,10,10}



● {75,10,10}

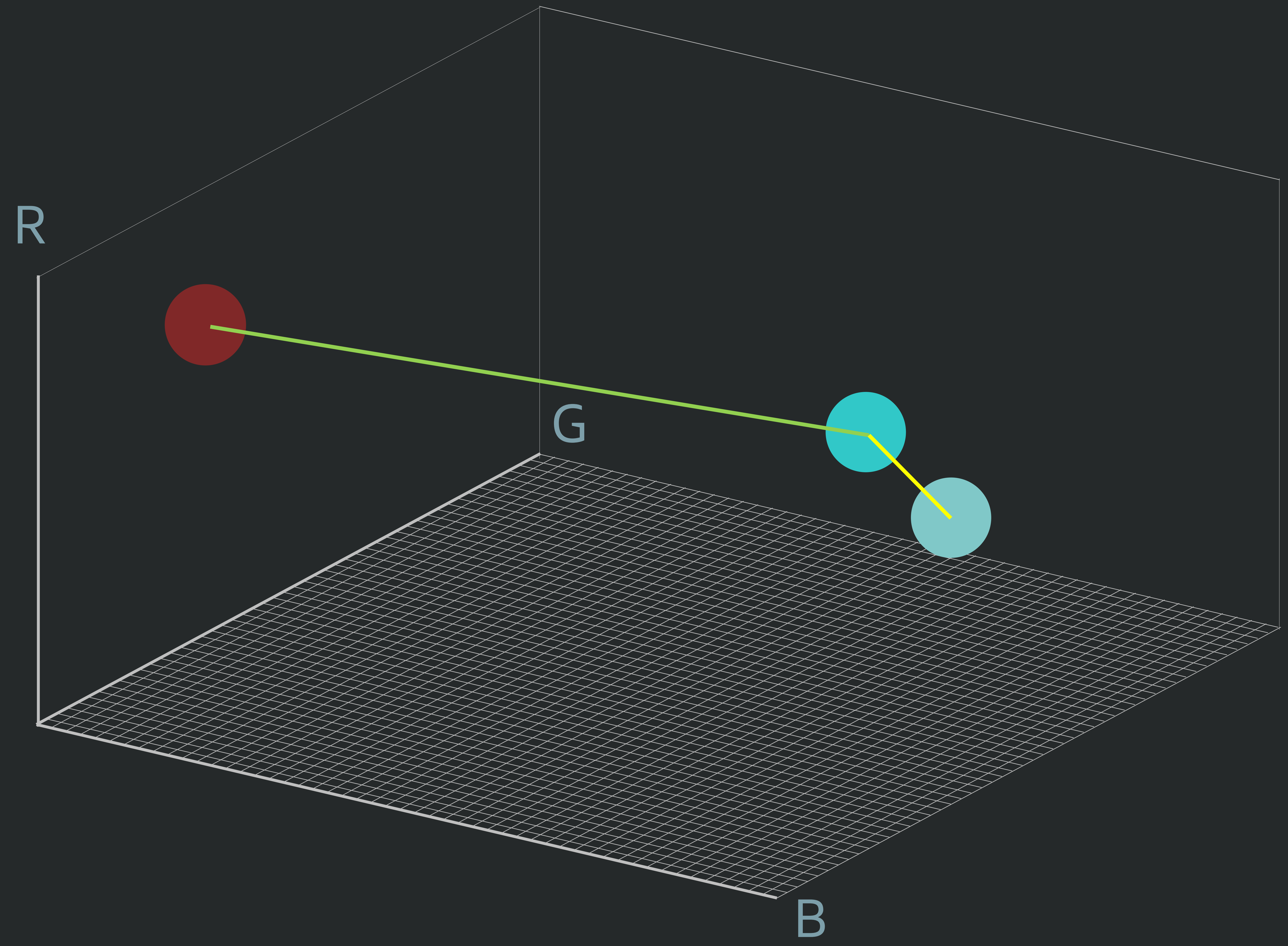
● {50,70,70}



● {75,10,10}

● {50,70,70}

● {25,75,75}

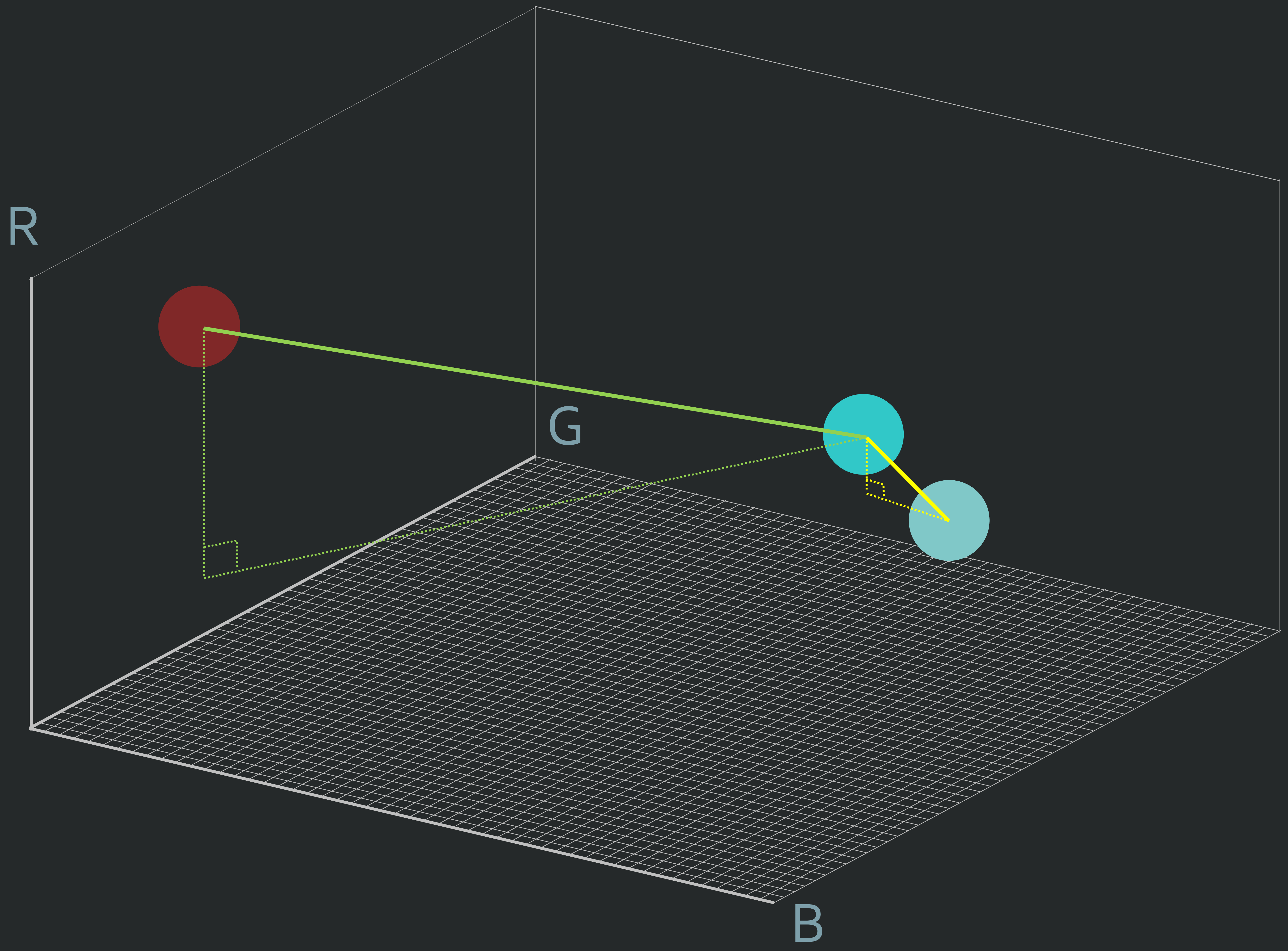


● {75,10,10}

● {50,70,70}

● {25,75,75}

distance = similarity

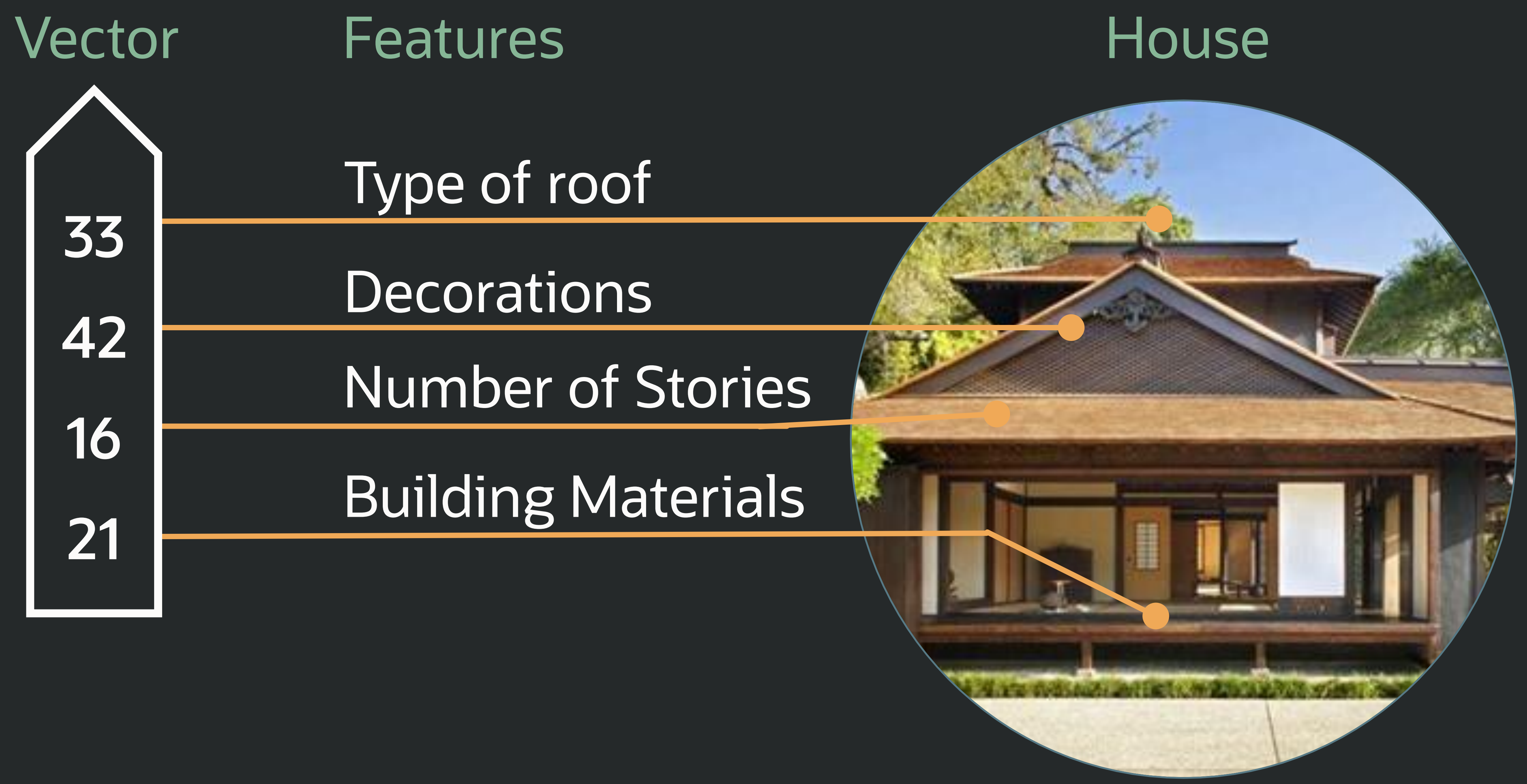


● {75,10,10}

● {50,70,70}

● {25,75,75}

more realistic example: house search



Vector

Features

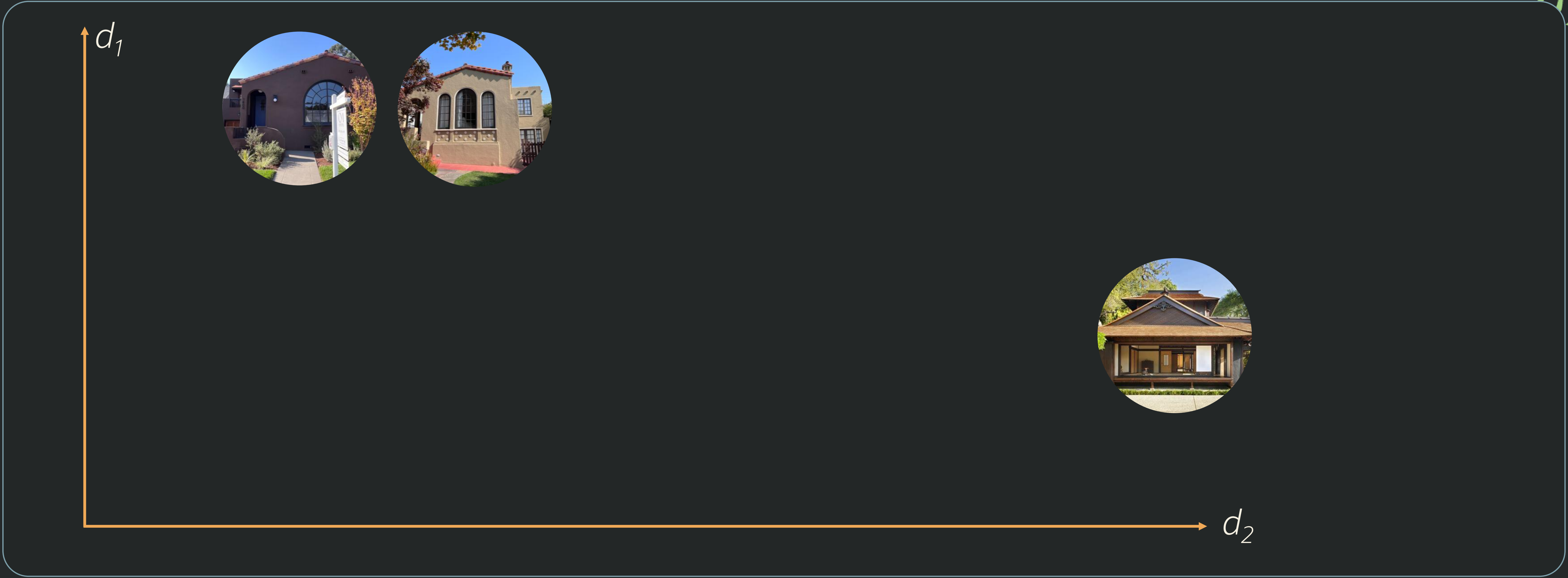
House

33 — Type of roof

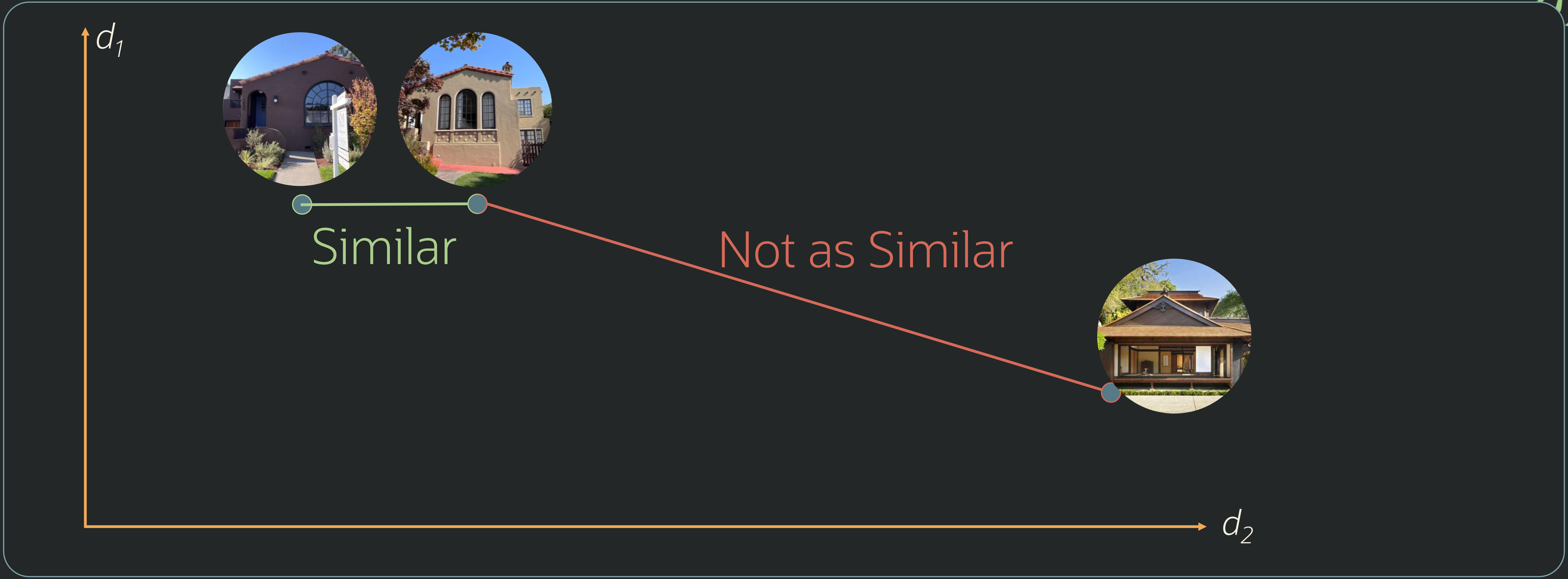
42 — Decorations

16 — Number of Stories

21 — Building Materials



(simplified to 2 dimensions)



distance defines similarity

can apply to anything

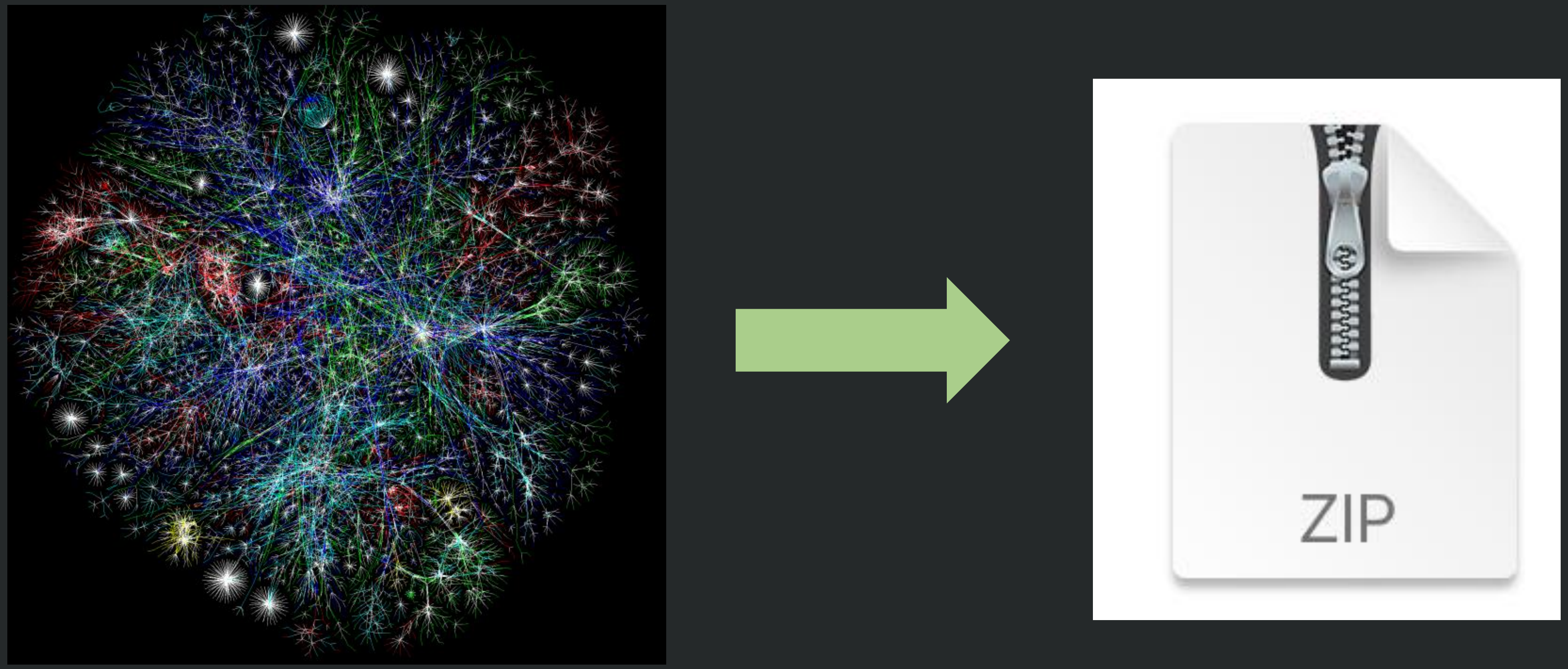
10TB housing data

10TB code

10TB medical studies

10TB literature

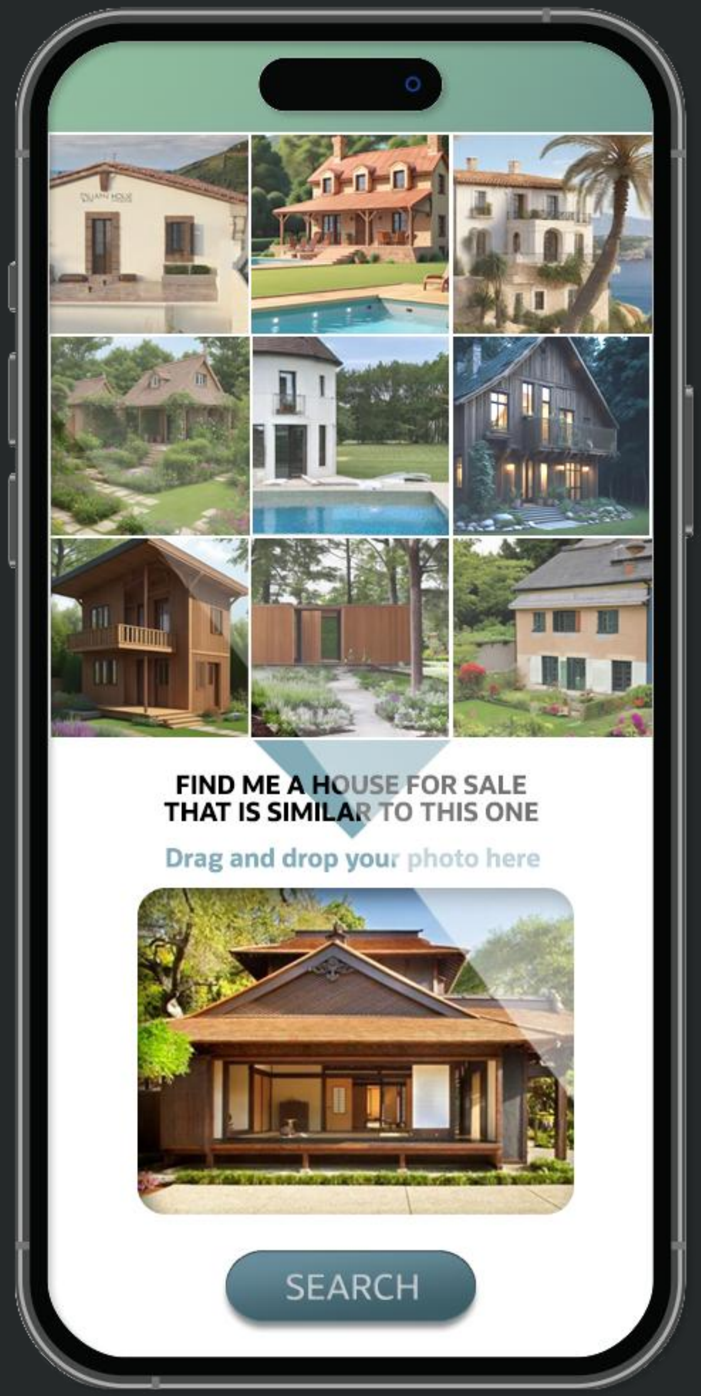
10TB "the internet"



Source: Intro to Large Language Models, Andrej Karpathy so many potential use cases

but we have a similar challenge

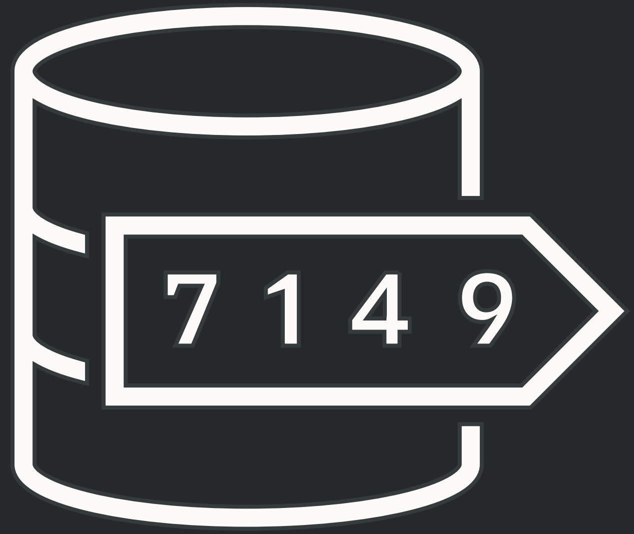
back to house hunting



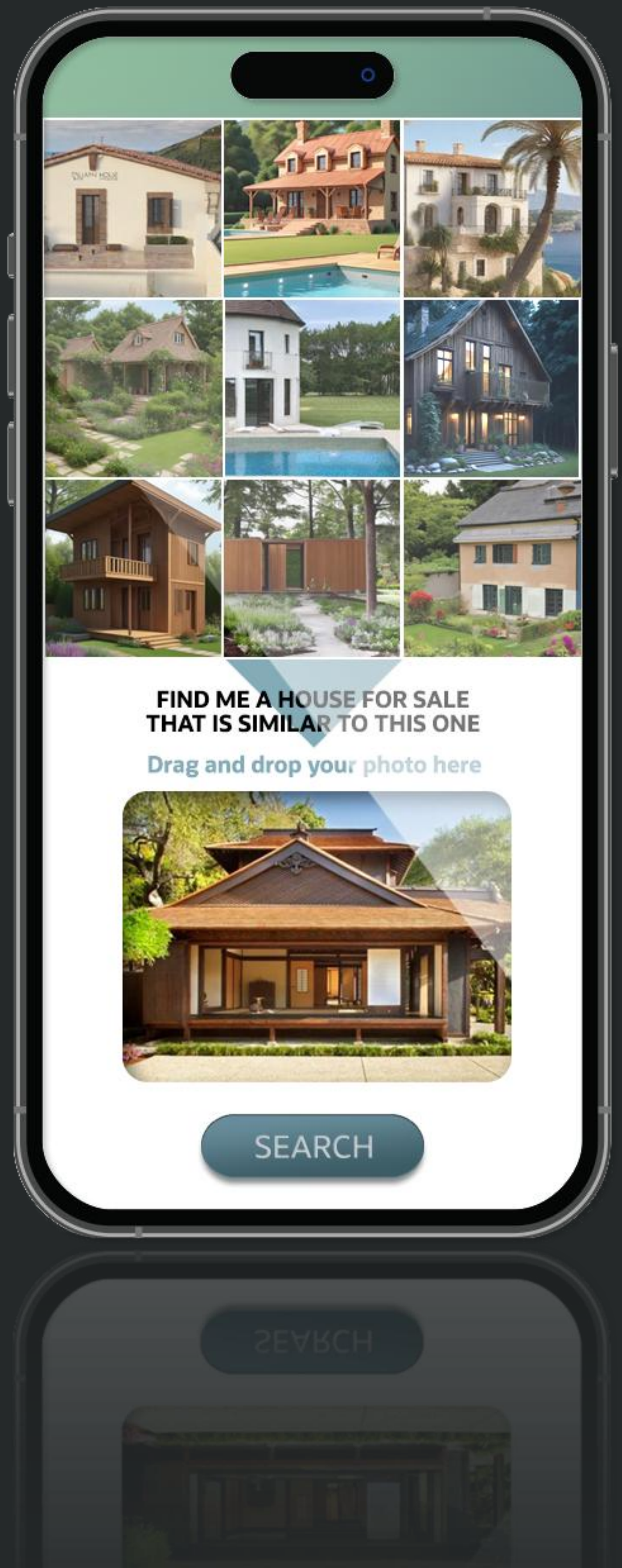
FIND ME A HOUSE FOR SALE
THAT IS SIMILAR TO THIS ONE

Drag and drop your photo here

start with image similarity search



Vector
Database

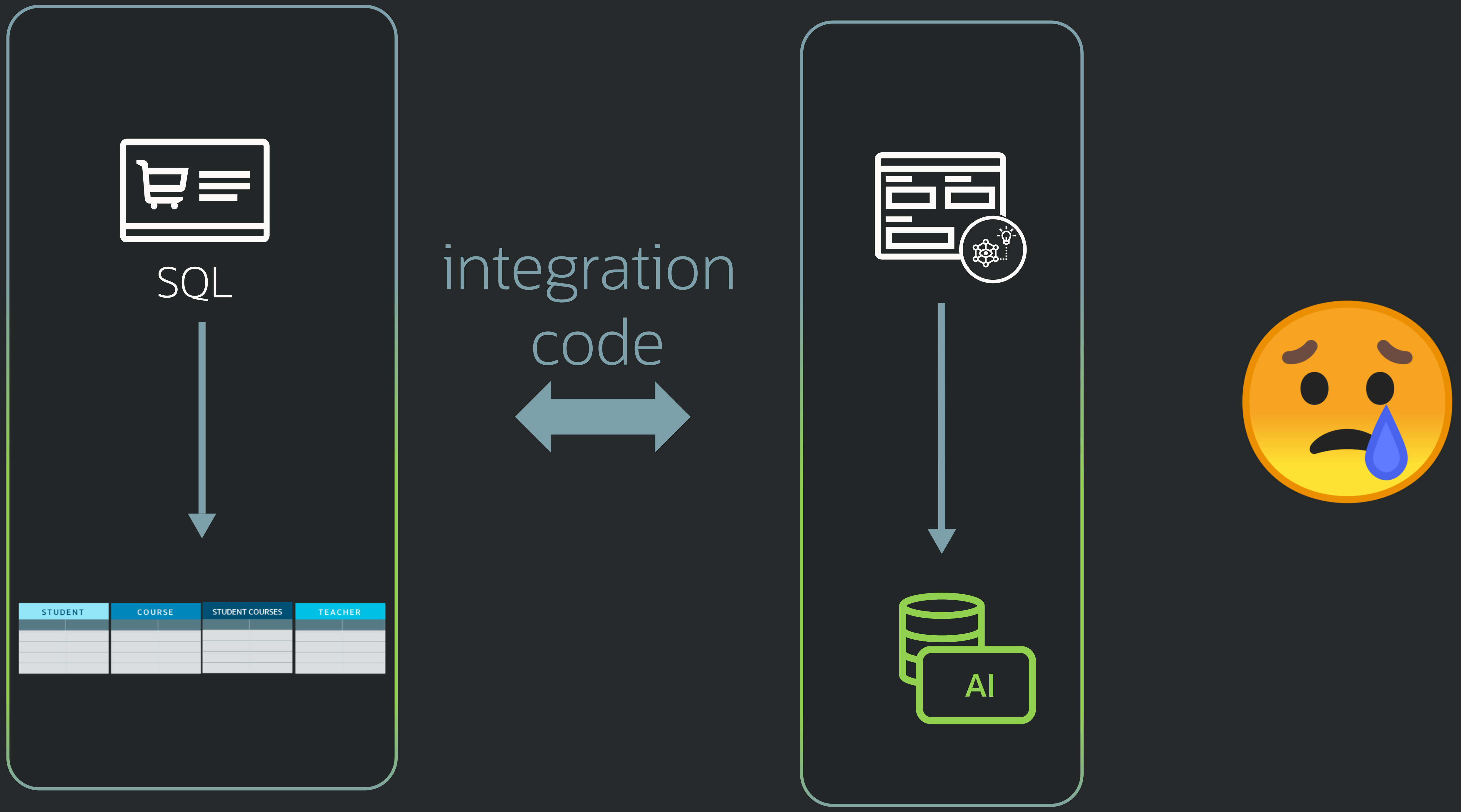


FIND ME A HOUSE FOR SALE
THAT IS SIMILAR TO THIS ONE

Drag and drop your photo here

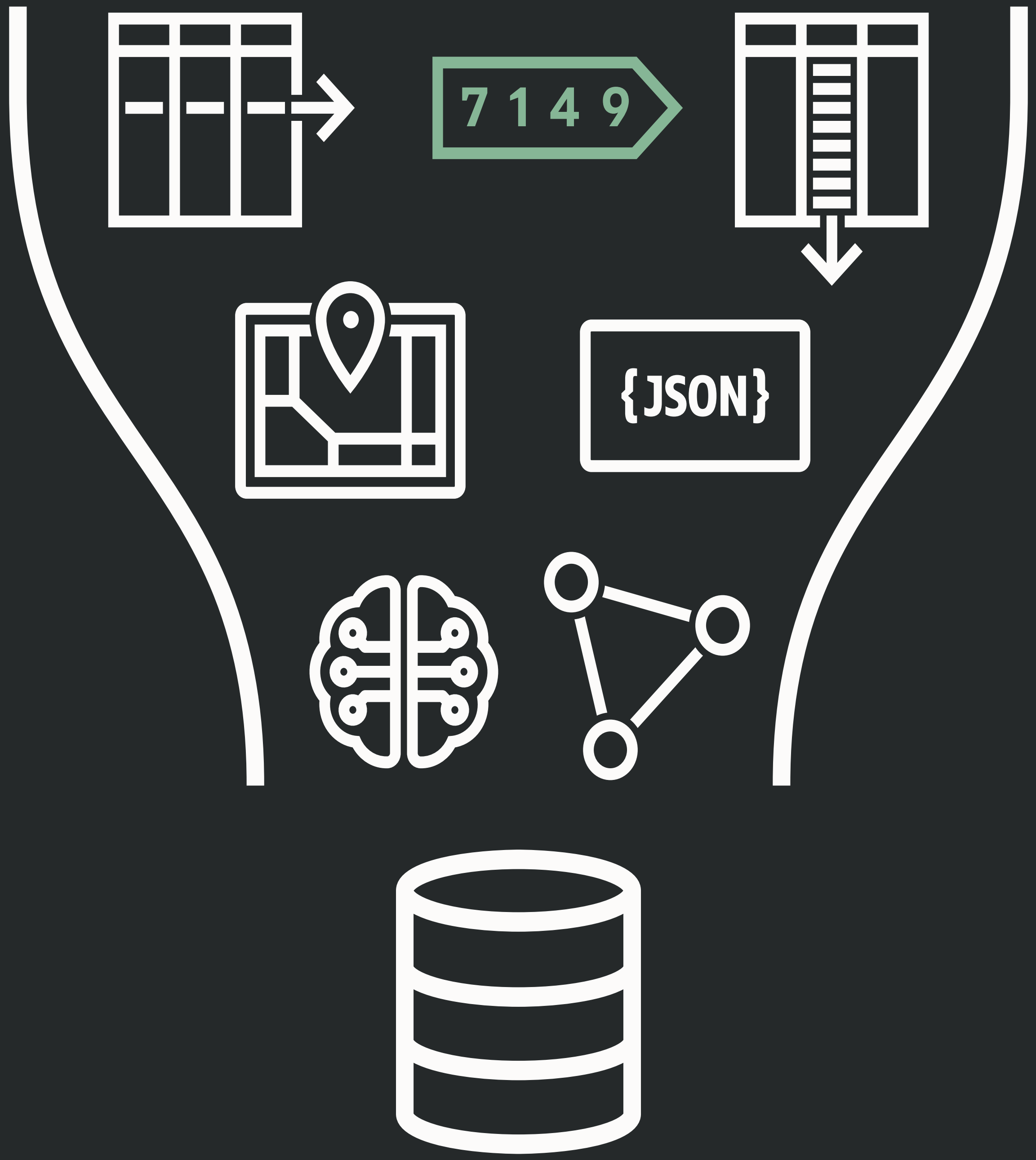
add customer budget, region prices, etc

Business
Database



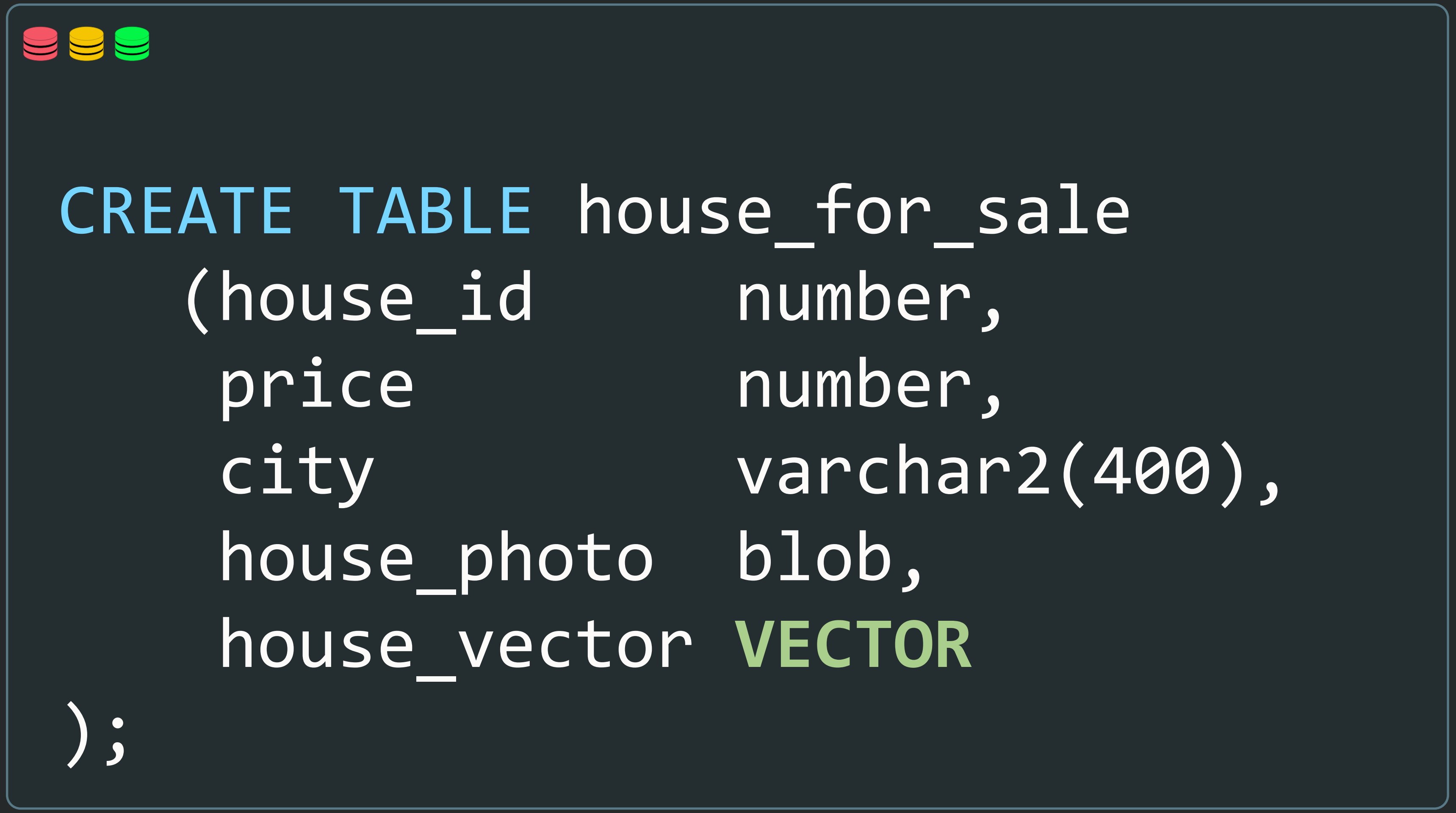
SQL

integration code



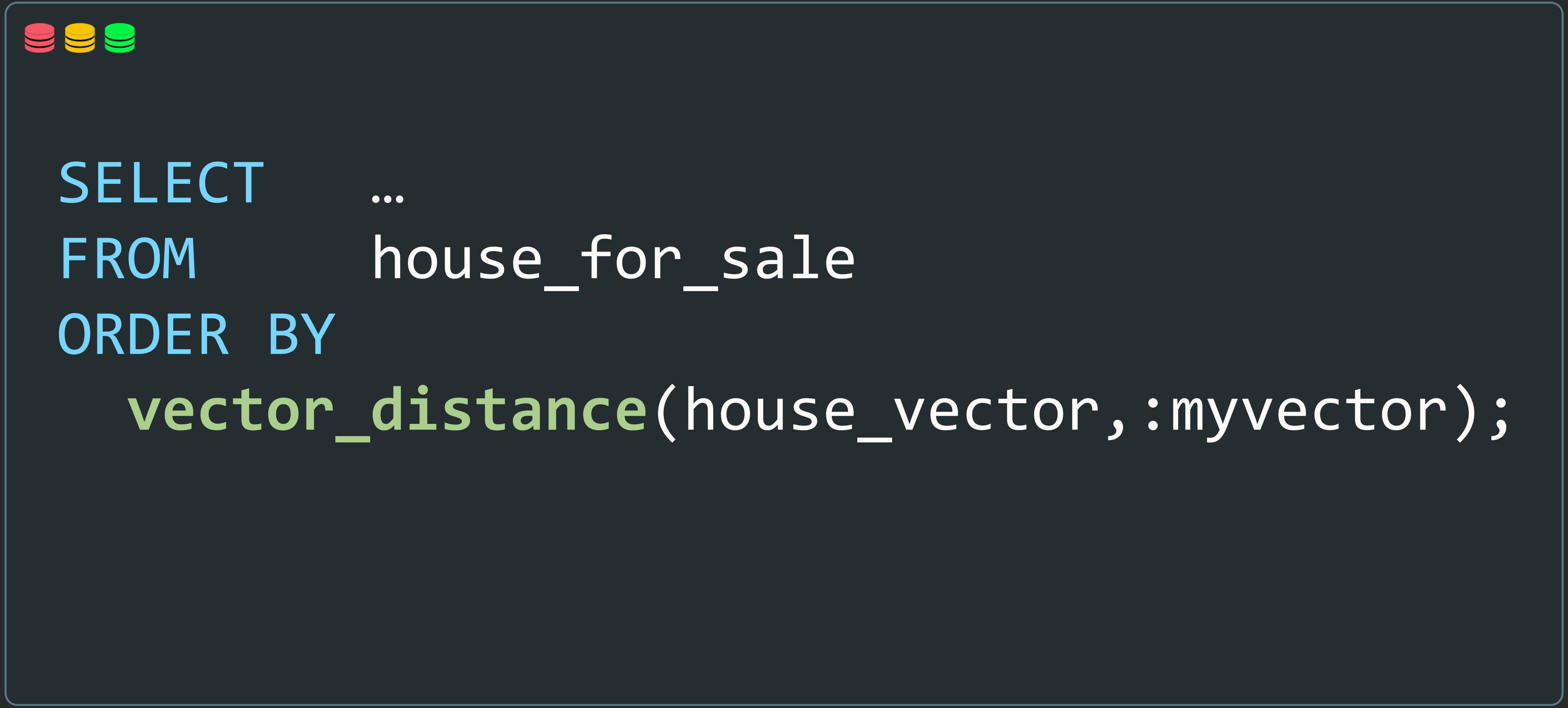
Converged Database

new datatype



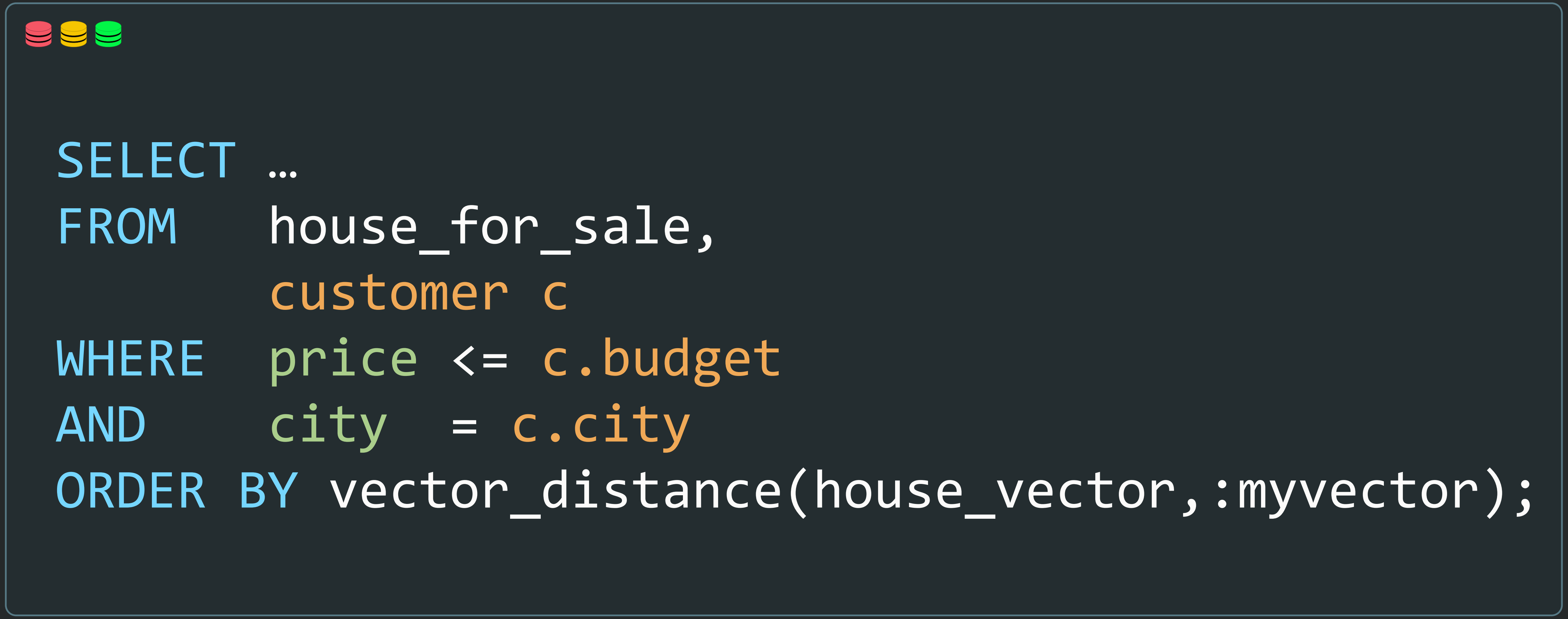
```
CREATE TABLE house_for_sale
   (house_id     number,
    price        number,
    city         varchar2(400),
    house_photo  blob,
    house_vector VECTOR
);
```

new functions



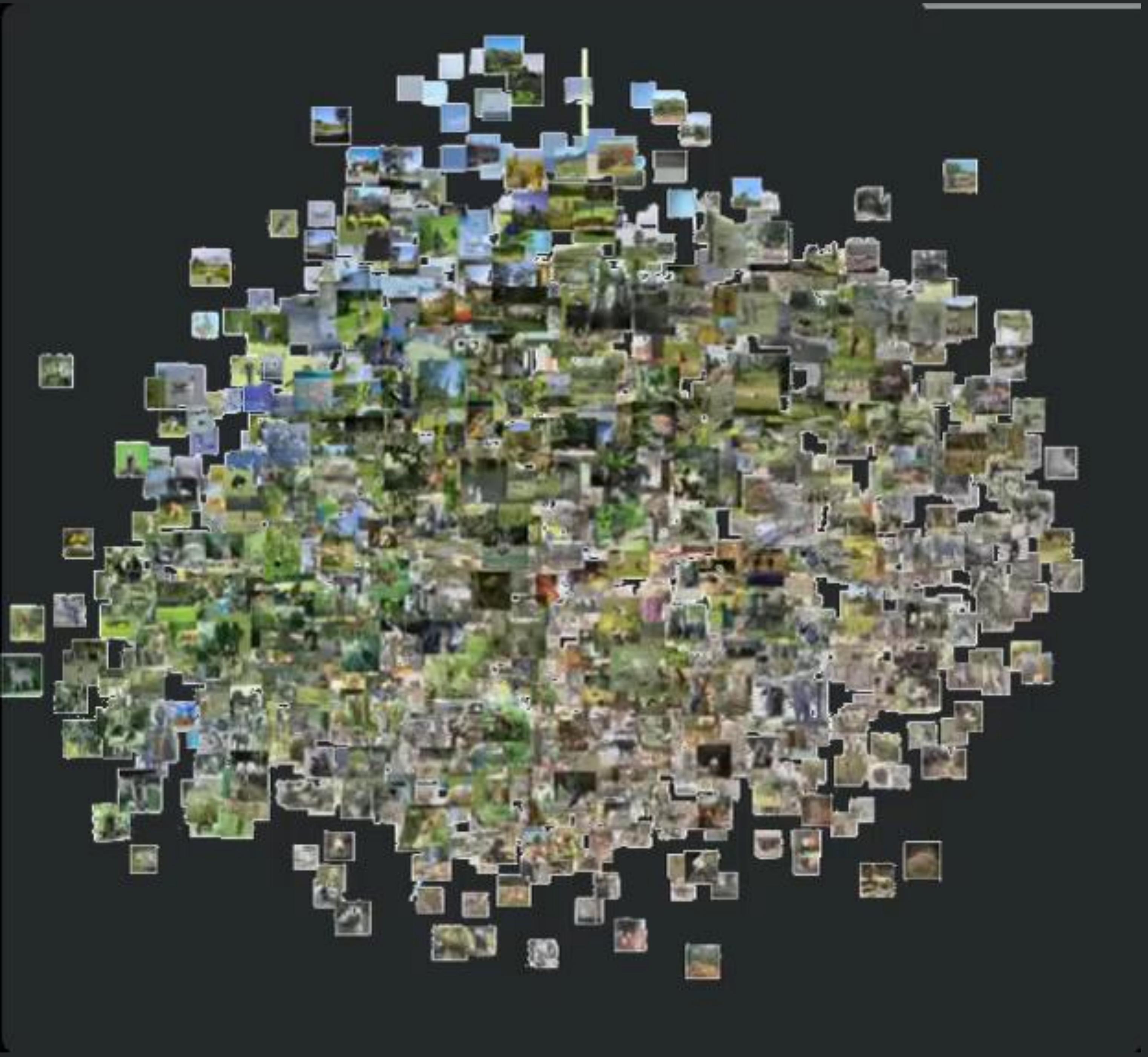
```
SELECT    …
FROM      house_for_sale
ORDER BY
  vector_distance(house_vector,:myvector);
```

converged



```
SELECT …
FROM   house_for_sale,
       customer c
WHERE  price <= c.budget
AND    city  = c.city
ORDER BY vector_distance(house_vector,:myvector);
```

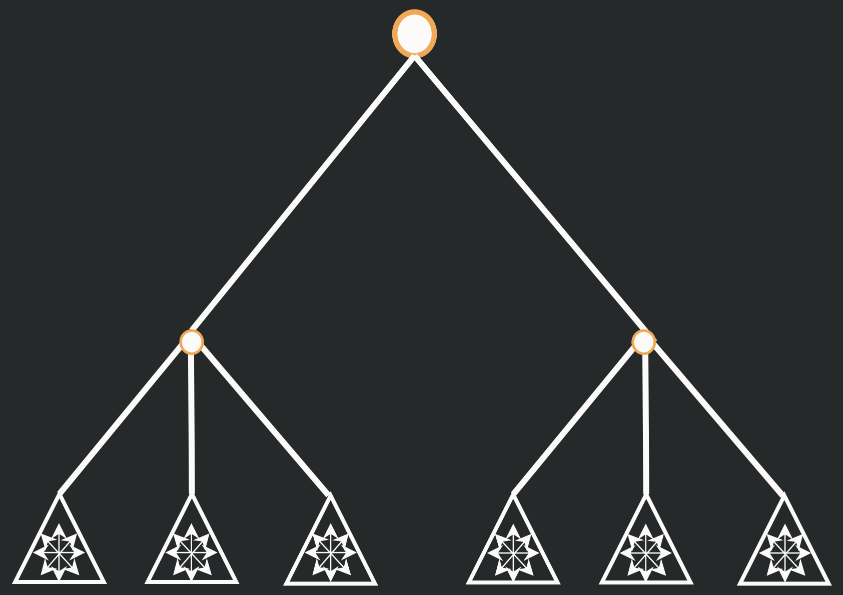
the challenge of vector search



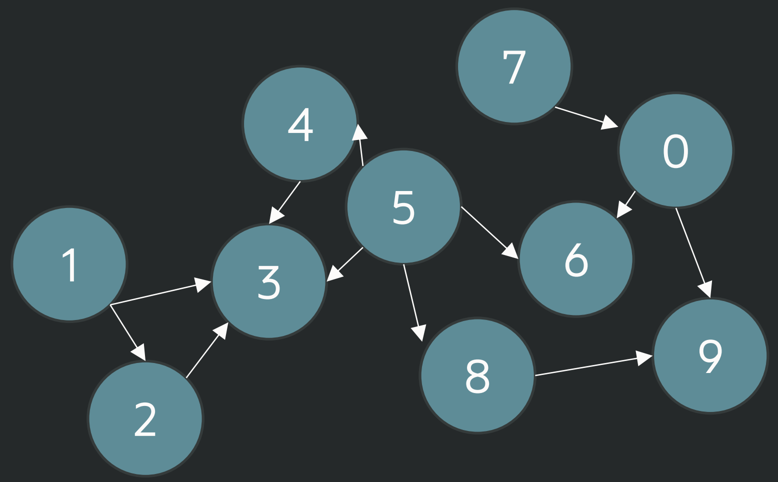
how to search this fast?

new vector indexes

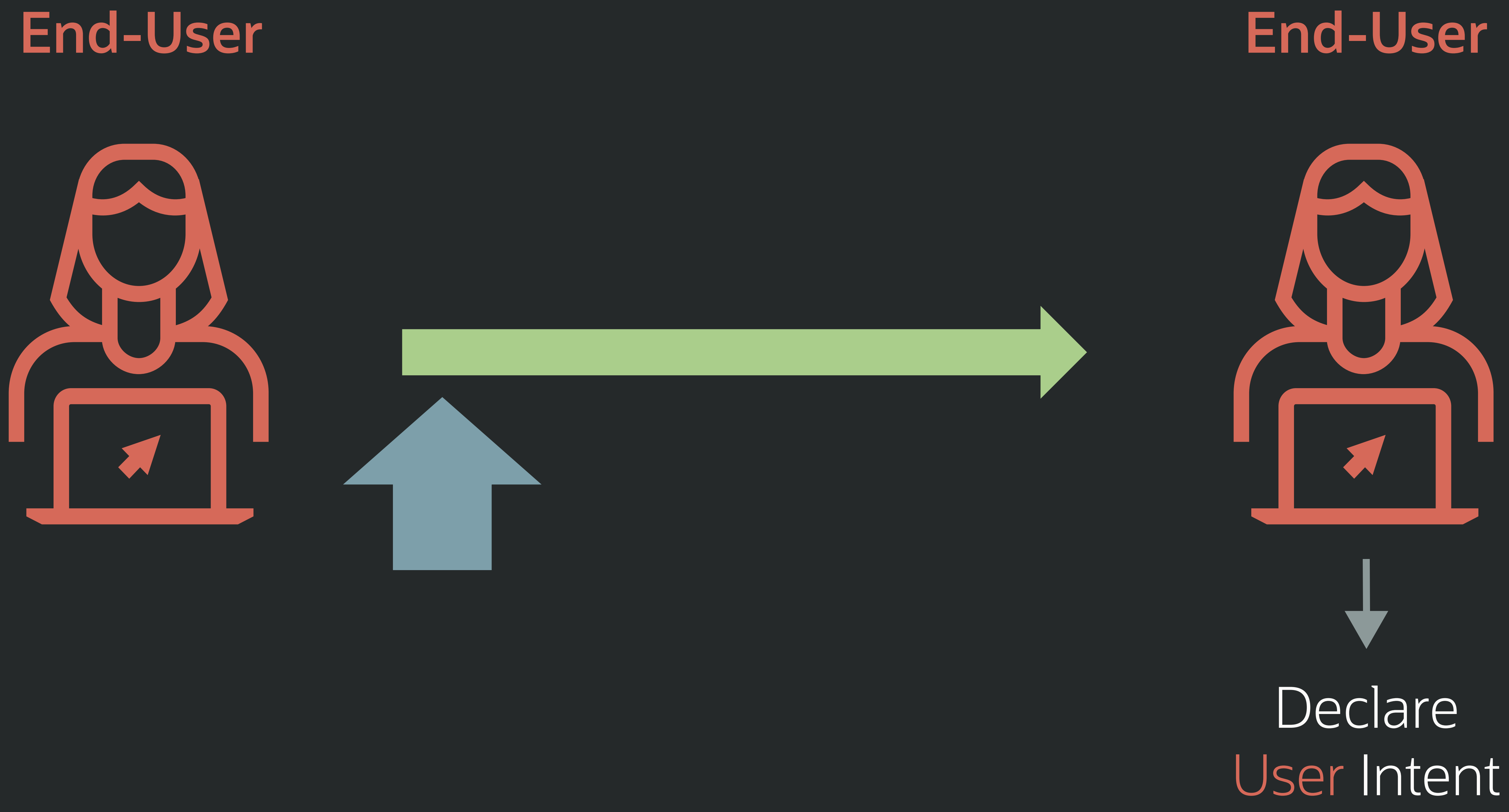
neighbourhood partitioned



neighbourhood graph



for the end user … we still aren't there



End-User

End-User

Declare
User Intent

natural language

SQL

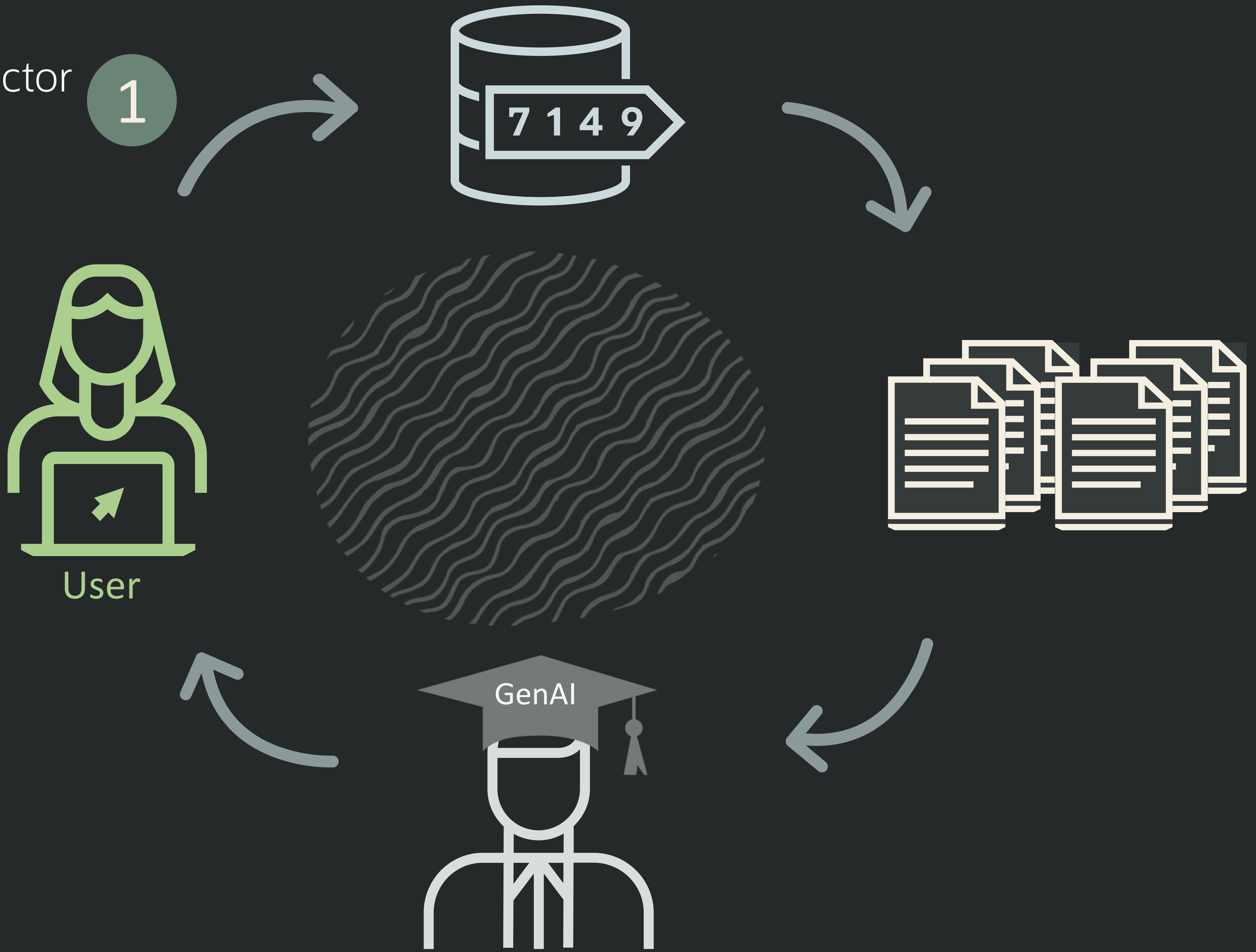
vector datatype / indexes

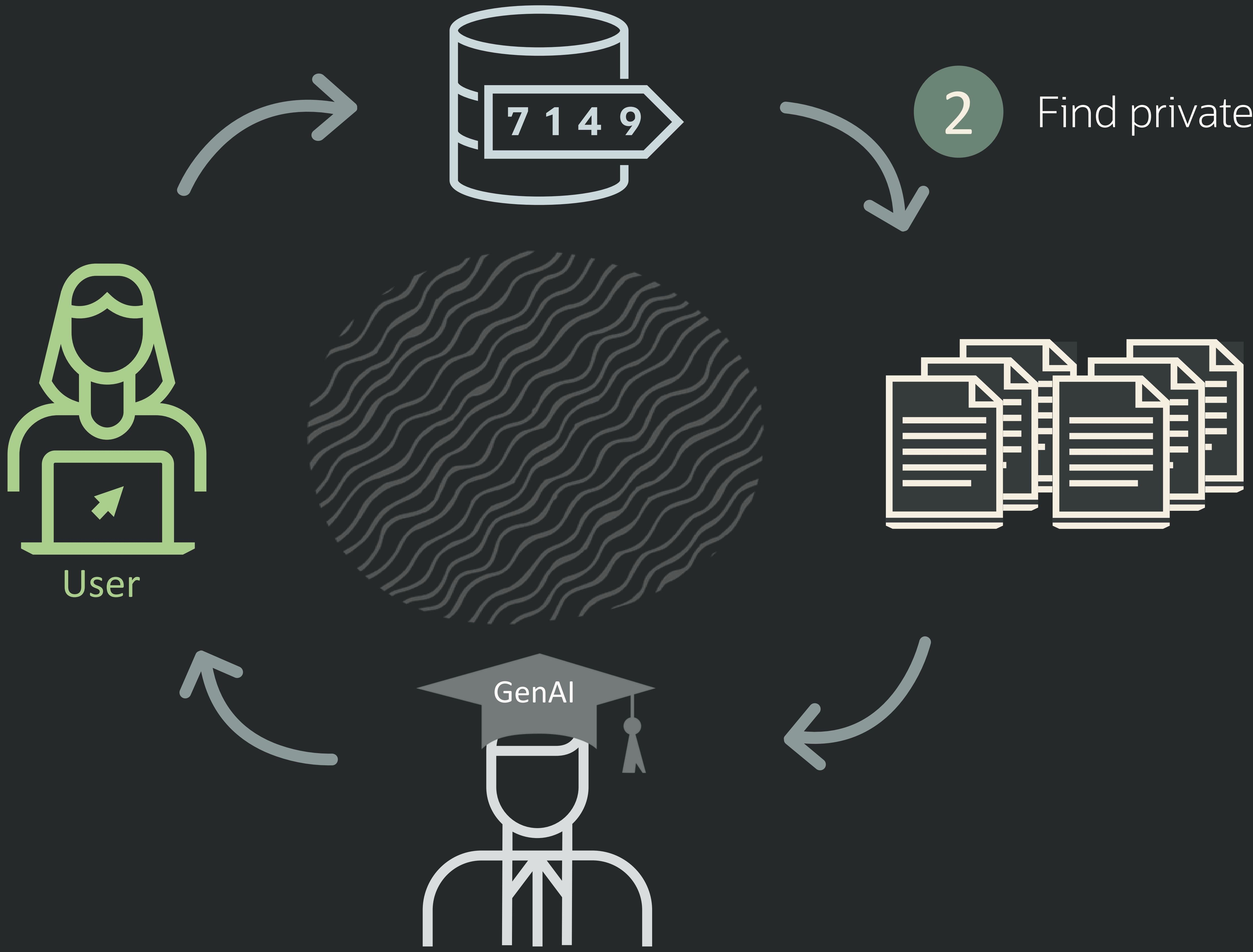
natural data

language, images, etc

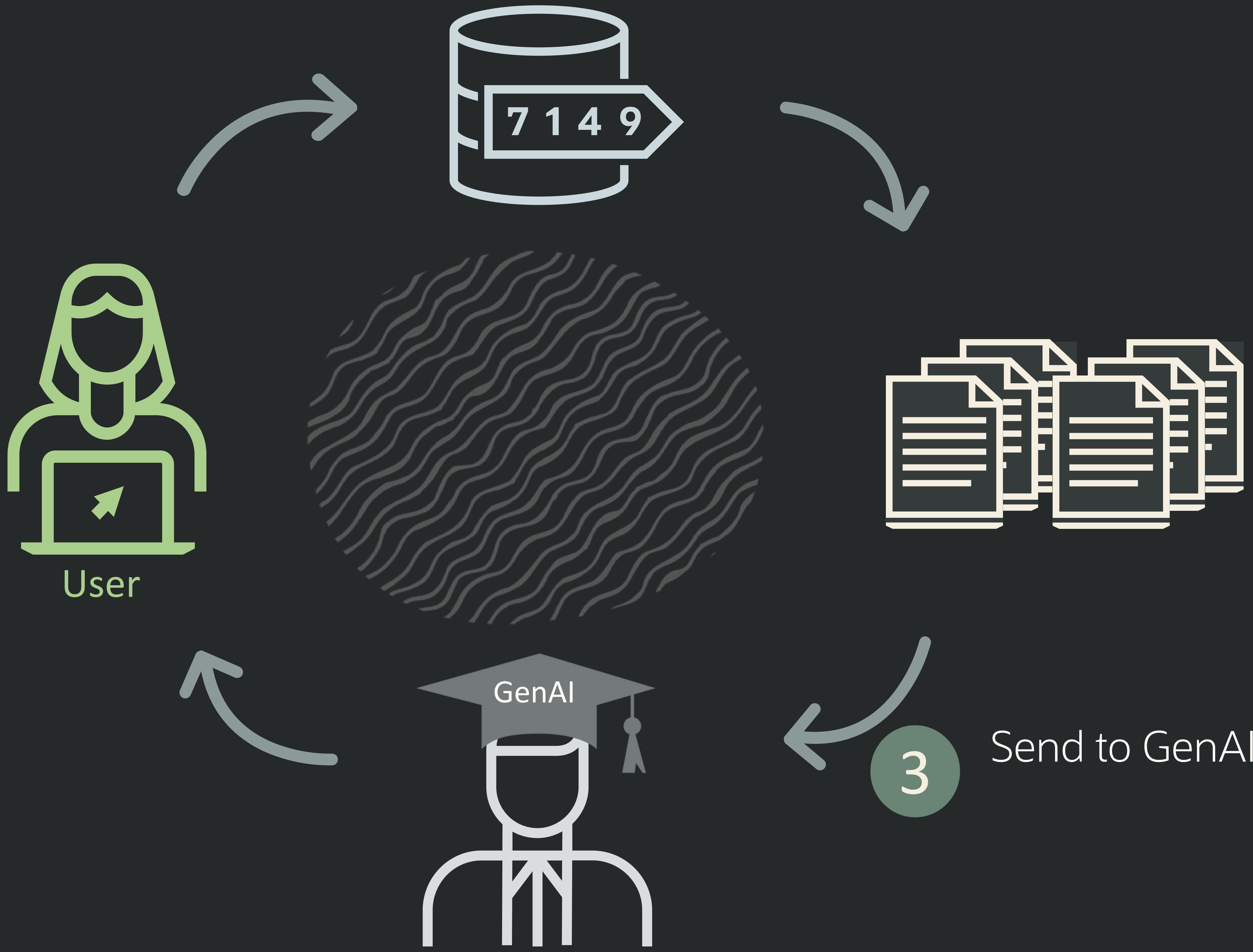
add Generative AI

Encode question as vector

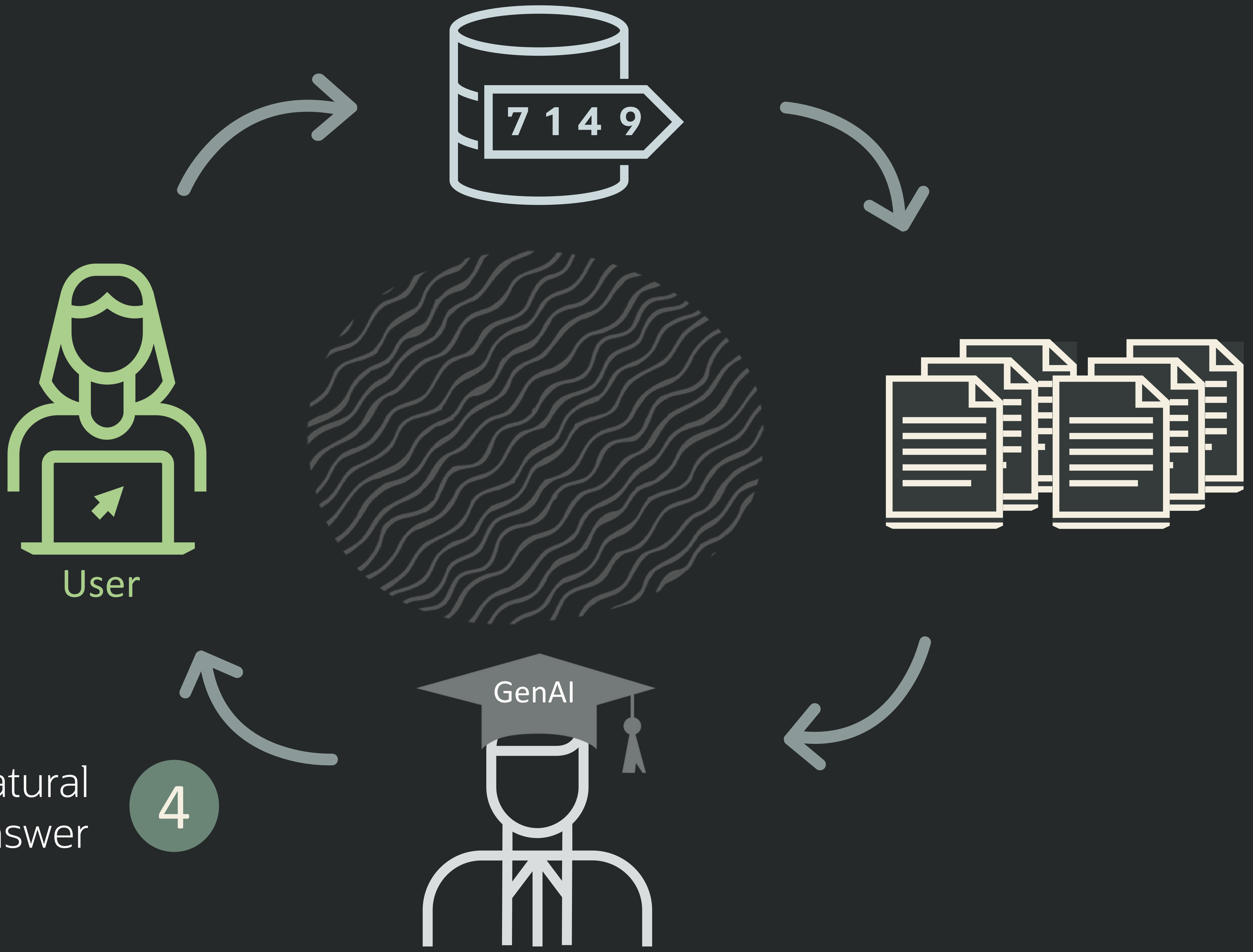




2 Find private data



3 Send to GenAI for text answer



4 GenAI provides natural language answer

what about privacy?

local
model

local
LLM

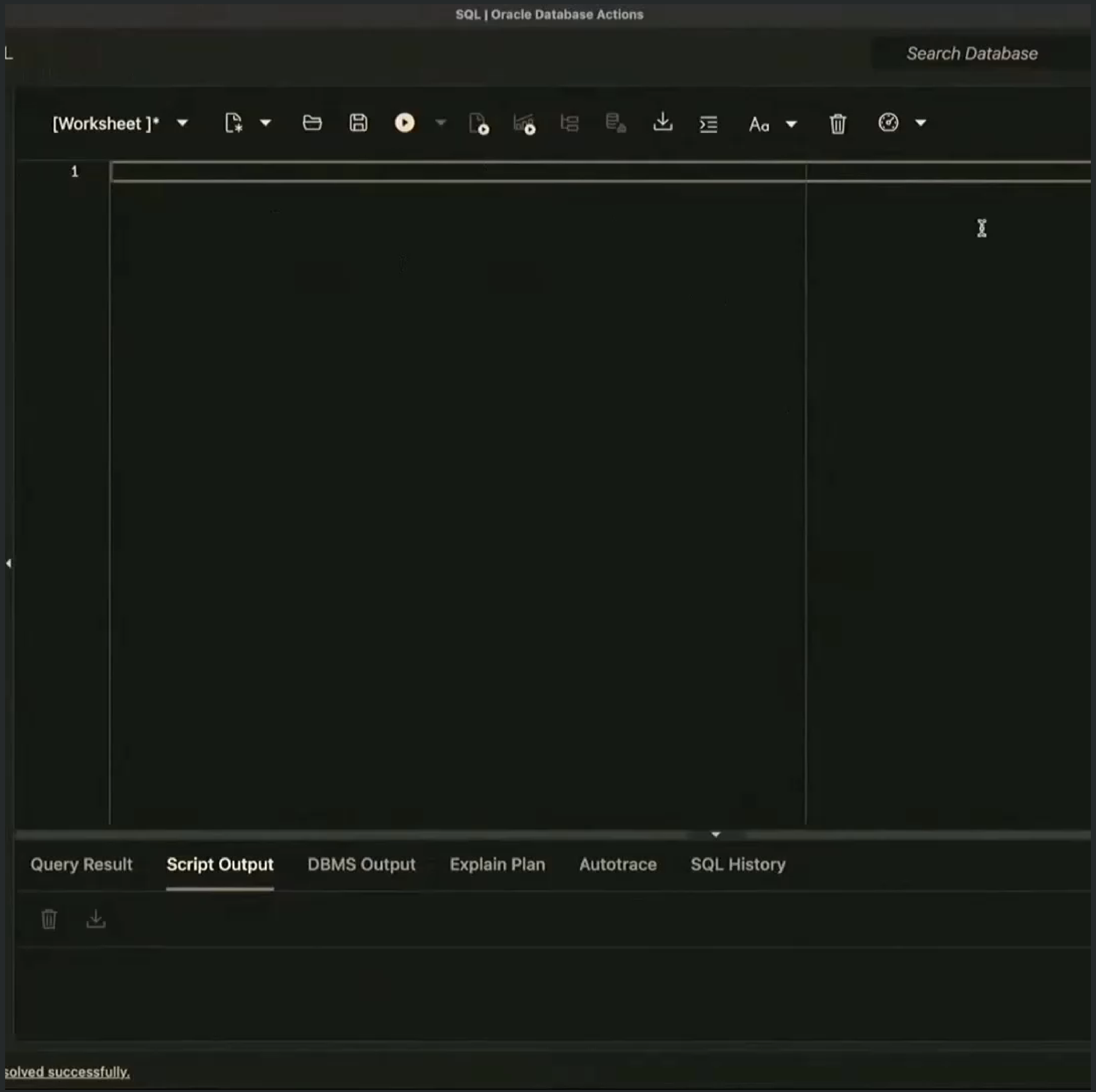
external
LLM

developers are people too ☺

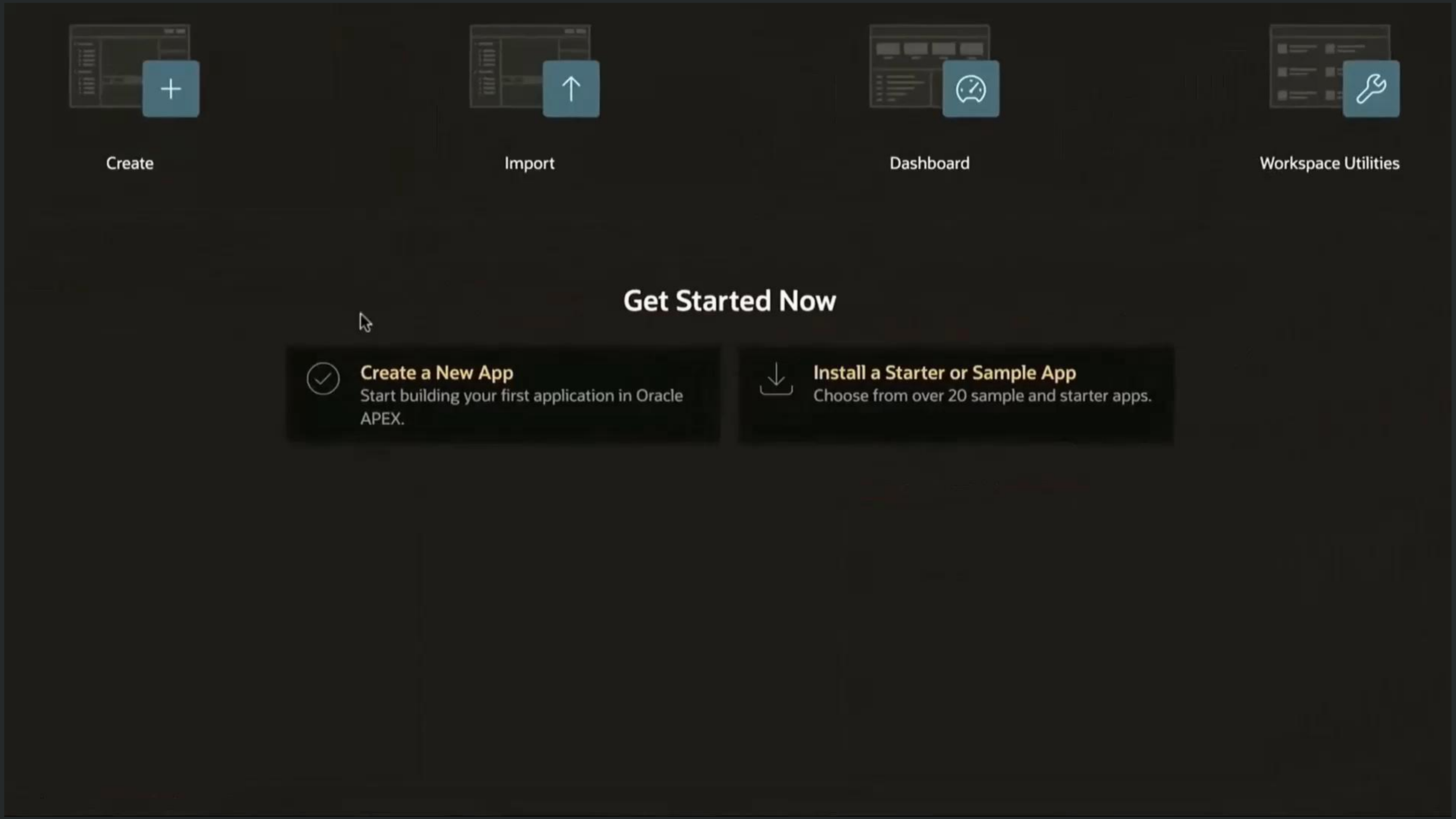
natural language as code

Developer

Declare
Data Usage Intent



data model intent



Get Started Now

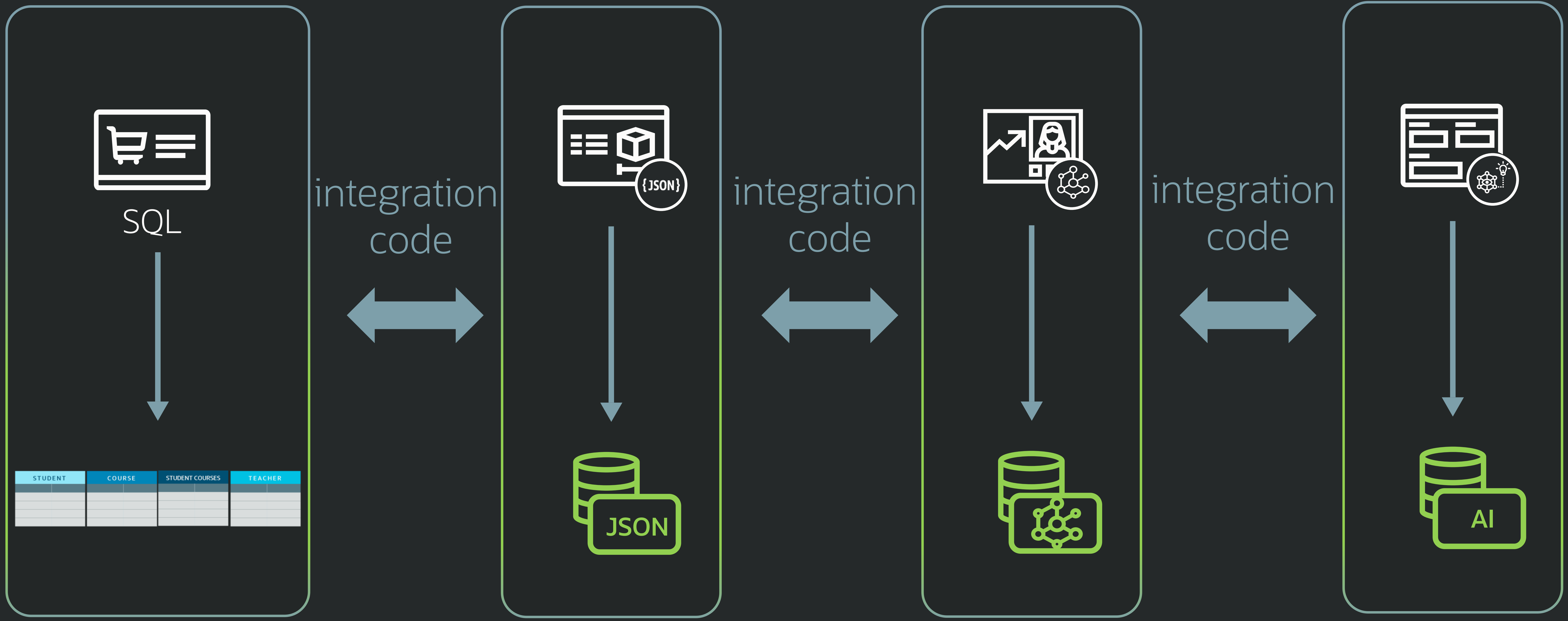
Create a New App
Start building your first application in Oracle APEX.

Install a Starter or Sample App
Choose from over 20 sample and starter apps.

coding intent

wrap up

we lost our way



SQL

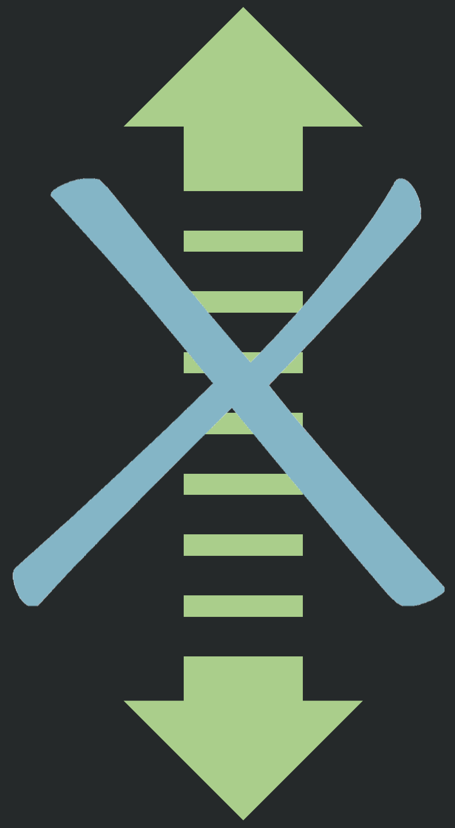
integration code

integration code

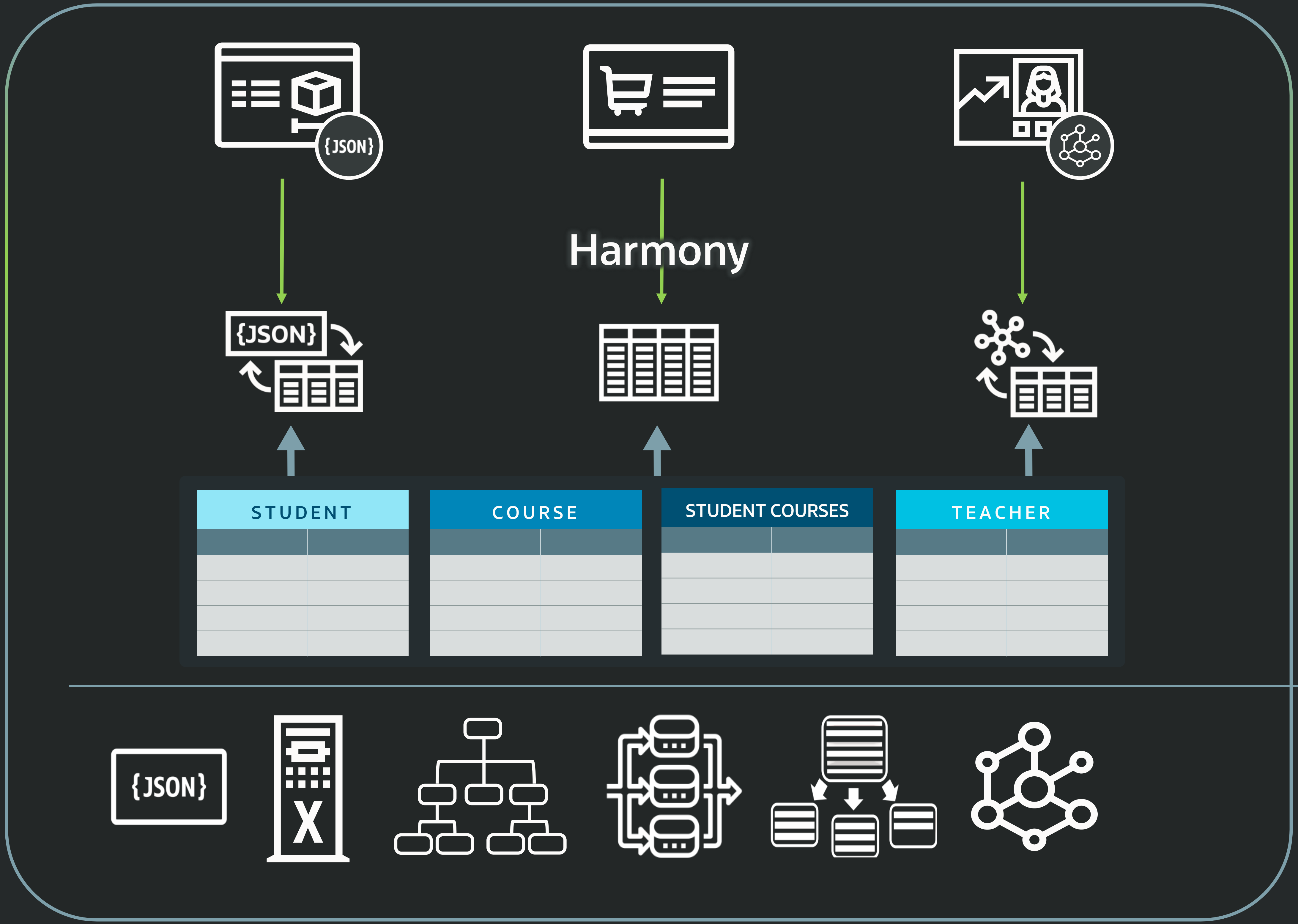
integration code

data usage = data storage

data usage



data storage

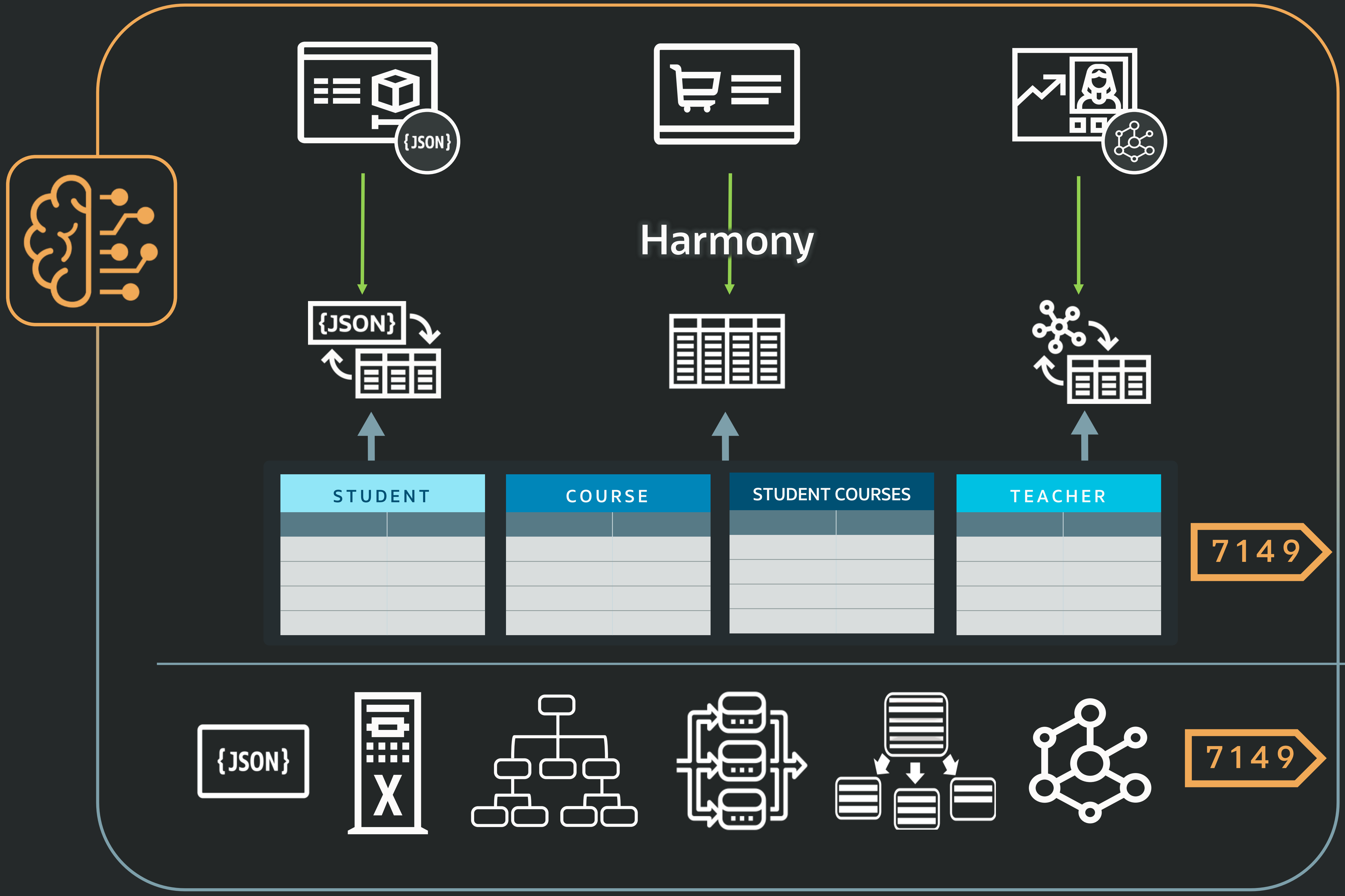


Database provides data as per usage needs

Simpler App Dev with none of the drawbacks

Relational Storage Model

Data consistency & quality



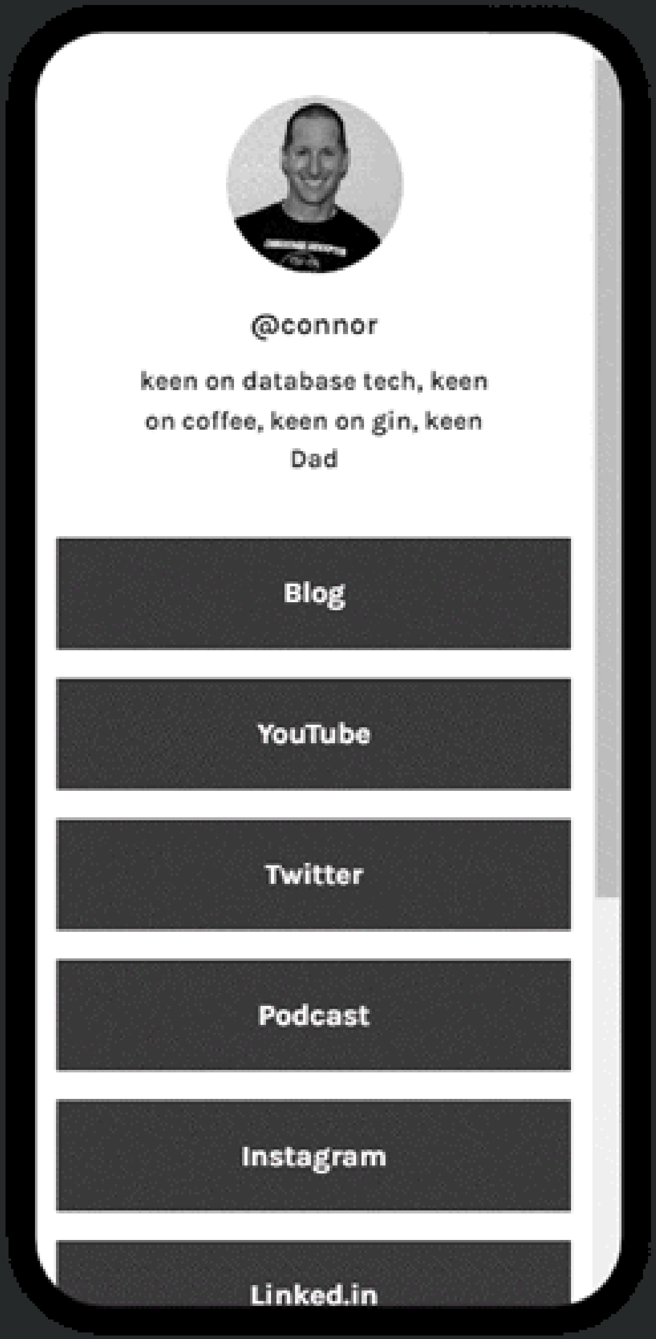
Generative AI
Allows natural language for questions, code and applications

AI Vectors bring semantic search to business data the future looks awesome!

Learn more

Session feedback

linktr.ee/connor



@connor

keen on database tech, keen on coffee, keen on gin, keen Dad