المقدمة

في هذا المشروع قمت بتحويل قاعدة بيانات تشغيلية من نوع OLTP إلى Data Warehouse بنظام الأبعاد (Dimensional Data Warehouse) لمشروع تأجير أفلام، باستخدام قاعدة بيانات sakila التي وفرتها .

قاعدة البيانات الأصلية OLTP كانت مصممة أساسا لتنفيذ العمليات اليومية مثل:

- تسجيل العملاء
- تأجير الأفلام
- عمليات الدفع
- إدارة المخزون
- إدارة الفروع
- متابعة الموظفين

لكن هذا النوع من القواعد يكون فيه الجداول كثيرة ومرتبطة ببعض بشكل معقد (Highly Normalized)، وهذا ممتاز للعمليات اليومية السريعة لكنه غير مناسب للتقارير والتحليلات واتخاذ القرارات.

هدف المشروع

الهدف الرئيسي كان تصميم Data Warehouse يسهل تحليل البيانات ويساعد المدراء ومحللي الأعمال على اتخاذ القرارات.

النظام الجديد يسمح بتحليل:

- اتجاهات التأجير
- سلوك العملاء
- أكثر الأفلام شعبية
- الإيرادات والأرباح
- أداء الفروع
- نشاط الموظفين

بشكل أسهل وأسرع من النظام التشغيلي الأصلي.

بداية العمل

في البداية قمتُ بتثبيت وإعداد:

- MySQL Server
- MySQL Workbench

ثم استوردت قاعدة البيانات الأصلية sakila بنجاح.

بعد دراسة مخطط قاعدة البيانات وفهم العمليات الأساسية، بدأت بتصميم نموذج الأبعاد باستخدام أسلوب:

Star Schema

وهو تصميم يحتوي على:

- جداول مركزية للحقائق (Fact Tables)
- محاطة بجداول الأبعاد (Dimension Tables)

حتى تصبح الاستعلامات التحليلية أبسط وأسرع.

تصميم الـ Data Warehouse

تم تنفيذ المخطط داخل قاعدة بيانات جديدة اسمها:

movie_rental_dw

وقمت بإنشاء جداول الأبعاد مثل:

- dim_customer
- dim_film
- dim_date
- dim_store
- dim_staff

وأيضًا جداول الحقائق مثل:

- fact_rental
- fact_payment

كما تم استخدام:

- Foreign Keys
- Surrogate Keys

للحفاظ على ترابط البيانات وتحسين الأداء.

رسم المخطط (ERD)

بعد إنشاء الجداول، استخدمت ميزة:

Reverse Engineer داخل MySQL Workbench

حتى أنشئ مخطط ERD يوضح:

- شكل الـ Star Schema
- العلاقات بين الجداول

بشكل بصري.

عملية ETL

في مرحلة ETL استخدمت:

- Python
- Pandas

- SQLAlchemy

داخل:

- VS Code
- Jupyter Notebook

وETL تعني:

1. Extract ← استخراج البيانات من قاعدة OLTP
2. Transform ← تعديل وتحويل البيانات لتناسب التحليل
3. Load ← تحميل البيانات داخل الـ Data Warehouse

ماذا فعلت في التحويل؟

قمت بـ:

- حساب مدة التأجير
- معرفة التأخير في الإرجاع
- دمج الاسم الأول والأخير للعملاء
- ربط الأفلام بتصنيفاتها
- تحميل البيانات المحولة داخل الـ DW

التحليلات والرسومات

أنشأت رسومات وتحليلات باستخدام:

- Pandas
- Matplotlib

لعرض أمثلة على التقارير مثل:

- تحليل إيرادات العملاء
- الإيرادات الشهرية
- تحليل فئات الأفلام

الأدوات المستخدمة

الأدوات والتقنيات المستخدمة في المشروع:

- MySQL Server
- MySQL Workbench
- Python
- Pandas
- SQLAlchemy
- Matplotlib
- VS Code / Jupyter Notebook

-

(Business Questions)

الهدف الأساسي من الـ Data Warehouse المقترح هو دعم التقارير التحليلية ومساعدة الإدارة على اتخاذ قرارات أفضل اعتماداً على البيانات التاريخية وبيانات العمليات اليومية.

بعد دراسة قاعدة بيانات الـ OLTP وفهم طريقة عمل شركة تأجير الأفلام، تم تحديد مجموعة من أسئلة الأعمال المهمة.

هذه الأسئلة تركز على تحليل:

- نشاط العملاء
- أداء عمليات التأجير
- الإيرادات
- شعبية الأفلام
- أداء الفروع
- التحليلات الزمنية

أولًا: أسئلة تحليل الأفلام (Film Analysis)

• ما هي الأفلام الأكثر استئجاراً؟

هذا السؤال يساعد في معرفة أكثر الأفلام شعبية عند العملاء.

الفائدة:

- تحسين إدارة المخزون
- توفير نسخ أكثر من الأفلام المطلوبة
- تحسين الحملات التسويقية

• ما هي الأفلام التي تحقق أعلى إيرادات؟

هذا التحليل يحدد الأفلام التي تجلب أكبر دخل للشركة.

الفائدة:

- **معرفة الأفلام الأكثر ربحية**
- **التركيز على الأفلام الناجحة مستقبلًا**

• ما هي فئات الأفلام الأكثر شعبية؟

مثل:

- **Action**
- **Comedy**
- **Drama**
- **Horror**

الفائدة:

- **فهم اهتمامات العملاء**
- **شراء أفلام جديدة حسب رغبة السوق**

ثانياً: أسئلة تحليل العملاء (Customer Analysis)

• من هم العملاء الأكثر نشاطاً في الاستئجار؟

أي العملاء الذين يستأجرون أفلاماً بكثرة.

الفائدة:

- **إنشاء برامج ولاء**
- **تقديم عروض خاصة للعملاء المهمين**

• أي العملاء يحققون أعلى إيرادات؟

ليس شرطًا أن يكون أكثرهم عددًا، لكن قد يدفعون أكثر.

الفائدة:

- **التركيز على العملاء ذوي القيمة العالية**
- **تحسين إدارة علاقات العملاء**

• كيف يتغير سلوك العملاء مع الوقت؟

تحليل نشاط العملاء عبر الأشهر أو السنوات.

الفائدة:

- معرفة المواسم النشطة
- فهم تفاعل العملاء مع الخدمة

ثالثاً: أسئلة تحليل الفروع (Store Analysis)

• أي فرع يحقق أكبر عدد من عمليات التأجير؟

الفائدة:

- تقييم أداء الفروع
- قياس الكفاءة التشغيلية

• أي فرع يحقق أعلى إيرادات؟

الفائدة:

- اتخاذ قرارات إدارية أفضل
- توزيع الموظفين والموارد بشكل مناسب

• كيف يختلف أداء الفروع حسب الموقع؟

تحليل الأداء حسب المنطقة الجغرافية.

الفائدة:

- معرفة الفروع القوية والضعيفة
- تحسين الخطط التجارية

رابعاً: التحليل الزمني (Time-Based Analysis)

• كيف يتغير نشاط التأجير شهرياً أو سنوياً؟

الفائدة:

- اكتشاف المواسم النشطة
- فهم اتجاهات العملاء

• ما هي اتجاهات الإيرادات الشهرية؟

متابعة الأرباح مع مرور الوقت.

الفائدة:

- تقييم نمو الشركة
- متابعة الأداء المالي

• ما الفترات التي تشهد أعلى طلب على التأجير؟

الفائدة:

- الاستعداد لفترات الضغط
- تحسين التخطيط والعروض

خامساً: تحليل الموظفين (Staff Analysis)

• أي الموظفين يعالج أكبر عدد من عمليات التأجير؟

الفائدة:

- قياس إنتاجية الموظفين
- توزيع عبء العمل

• أي الموظفين يتعامل مع أكبر عدد من عمليات الدفع؟

الفائدة:

- مراقبة الأداء والكفاءة

سادساً: تحليل مدة التأجير والإرجاع (Rental Duration & Return Analysis)

• ما متوسط مدة استئجار الأفلام؟

الفائدة:

- فهم عادات العملاء
- معرفة مدة استخدام الأفلام

• ما أكثر عمليات التأجير التي يحدث فيها تأخير بالإرجاع؟

الفائدة:

- تحسين سياسات الغرامات
- تقليل التأخير

• أي فئات الأفلام يحدث فيها تأخير أكثر؟

الفائدة:

- اكتشاف أنماط سلوك العملاء
- فهم تأثير نوع الفيلم على التأخير

تصميم النموذج البُعدي (Dimensional Model Design)

- ستجد كود sql في ملف منفصل

تم تصميم النموذج البُعدي اعتمادًا على المتطلبات التحليلية الأساسية لنظام تأجير الأفلام.

بعد تحليل قاعدة بيانات الـ OLTP وفهم العمليات المهمة داخل النظام، تم اختيار تصميم:

Star Schema

لأنه:

- يبسط الاستعلامات التحليلية
- يحسن أداء التقارير
- يجعل فهم البيانات أسهل

النموذج المصمم يحتوي على:

- جداول حقائق (Fact Tables) في المنتصف
- مرتبطة بجداول أبعاد (Dimension Tables)

تصف:

- العملاء
- الأفلام
- الفروع
- الموظفين
- الوقت والتواريخ

العمليات التجارية الأساسية المختارة

النظام يركز على عمليتين رئيسيتين:

- عمليات تأجير الأفلام (Movie Rental Transactions)
- عمليات الدفع (Payment Transactions)

جداول الحقائق (Fact Tables)

جداول الحقائق تخزن البيانات الرقمية والعمليات التي يمكن تحليلها.

أولًا: جدول fact_rental

العملية التجارية

هذا الجدول يخزن جميع عمليات تأجير الأفلام التي يقوم بها العملاء.

الـ Grain (مستوى التفاصيل)

كل صف داخل الجدول يمثل:

عملية تأجير واحدة
لعميل واحد
لفيلم واحد
بتاريخ محدد
وفي فرع محدد

يعني:
كل عملية تأجير = سجل مستقل.

المقاييس (Measures)

rental_duration_days

عدد أيام التأجير.

is_late_return

هل تم إرجاع الفيلم متأخرًا أم لا.

الأبعاد المرتبطة

الجدول مرتبط مع:

- dim_date
- dim_customer
- dim_film
- dim_store
- dim_staff

الهدف من الجدول

يستخدم لتحليل:

- نشاط التأجير
- سلوك العملاء
- شعبية الأفلام
- التأخير في الإرجاع

- آداء الفروع

ثانيًا: جدول fact_payment

العملية التجارية

هذا الجدول يخزن جميع عمليات الدفع المرتبطة بالتأجير.

الـ Grain

كل صف يمثل:

عملية دفع واحدة قام بها عميل واحد.

المقاييس

amount

قيمة الدفع أو المبلغ المالي.

الأبعاد المرتبطة

مرتبط مع:

- dim_date
- dim_customer
- dim_staff
- fact_rental

الهدف من الجدول

يدعم:

- تحليل الإيرادات
- تحليل إنفاق العملاء
- التقارير المالية
- تتبع عمليات الدفع

جداول الأبعاد (Dimension Tables)

جداول الأبعاد تخزن المعلومات الوصفية التي تساعد في:

- التصفية
- التجميع
- التحليل

جدول dim_date

الهدف

دعم التحليل الزمني والتقارير.

أهم الحقول

- date_key
- full_date
- day_name
- month_number
- month_name
- quarter
- year

مصدر البيانات في OLTP

تم أخذ البيانات من:

- rental
- payment

الاستخدام التحليلي

يستخدم في:

- التحليل الشهري
- التحليل السنوي
- التقارير الموسمية

جدول dim_customer

الهدف

تخزين معلومات العملاء لتحليل سلوكهم.

أهم الحقول

- customer_id
- first_name
- last_name
- full_name
- email
- active
- create_date

مصدر البيانات

من جدول:

- customer

الاستخدام التحليلي

يستخدم في:

- تحليل نشاط العملاء
- الإيرادات حسب العميل
- تقسيم العملاء لفئات

جدول dim_film

الهدف

تخزين المعلومات الوصفية الخاصة بالأفلام.

أهم الحقول

- film_id
- title
- description
- release_year
- rental_duration
- rental_rate
- length_minutes
- replacement_cost
- rating
- category_name
- language_name

مصدر البيانات

من الجداول:

- film
- category
- film_category
- language

الاستخدام التحليلي

يستخدم في:

- تحليل شعبية الأفلام
- تحليل التصنيفات
- الإيرادات حسب الفيلم

جدول dim_store

الهدف

تخزين معلومات الفروع والمواقع.

أهم الحقول

- store_id
- manager_first_name
- manager_last_name
- city
- country

مصدر البيانات

من:

- store
- address
- city
- country
- staff

الاستخدام التحليلي

يستخدم في:

- تحليل أداء الفروع
- الإيرادات حسب الموقع
- التقارير الإقليمية

جدول dim_staff

الهدف

تخزين معلومات الموظفين لتحليل الأداء.

أهم الحقول

- staff_id
- first_name

- last_name
- email

مصدر البيانات

من جدول:

- staff

الاستخدام التحليلي

يستخدم في:

- تحليل إنتاجية الموظفين
- تحليل عمليات التأجير
- متابعة نشاط الدفع

الـ Grain

الـ Grain يعني:

أقل مستوى من التفاصيل داخل جدول الحقائق.

Grain في fact_rental

كل صف يمثل:

عملية تأجير واحدة
قام بها عميل واحد
لفيلم واحد
بتاريخ محدد.

Grain في fact_payment

كل صف يمثل:

عملية دفع واحدة
قام بها عميل واحد.

لماذا تحديد الـ Grain مهم؟

لأنه:

- يمنع تكرار البيانات
- يحافظ على دقة التقارير
- يجعل التحليل متناسقًا

المقاييس (Measures)

المقاييس هي القيم الرقمية داخل جداول الحقائق والتي تَستخدم في العمليات الحسابية والتحليل.

مقاييس fact_rental

- rental_duration_days
- is_late_return

ماذا نستفيد منها؟

تساعد في:

- تحليل مدة التأجير
- اكتشاف التأخير
- فهم سلوك العملاء

مقاييس fact_payment

- amount

ماذا نستفيد منه؟

يدعم:

- تحليل الإيرادات
- التقارير المالية
- تحليل اتجاهات الدفع

مخطط الـ Star Schema

تم تنفيذ النموذج البُعدي باستخدام تصميم:

Star Schema

لأنه يوفر:

- **بنية بسيطة**
- **سرعة أكبر في التحليل**
- **سهولة في كتابة التقارير والاستعلامات**

كيف يعمل الـ Star Schema؟

في هذا التصميم:

- جداول الحقائق (Fact Tables) تكون في المنتصف
- وجداول الأبعاد (Dimension Tables) تحيط بها

وترتبط معها باستخدام:

- Foreign Keys

جداول الحقائق الرئيسية

النظام يحتوي على جدولين أساسيين:

- fact_rental
- fact_payment

جداول الأبعاد المرتبطة

هذه الجداول مرتبطة بجداول الحقائق:

- dim_date
- dim_customer
- dim_film
- dim_store
- dim_staff

لماذا الـ Star Schema مهم؟

لأنه:

- يقلل تعقيد الـ JOIN
- يجعل التحليل أسرع
- أسهل من قاعدة الـ OLTP الأصلية

في الـ OLTP كان هناك عدد كبير من الجداول والعلاقات، أما هنا فالتحليل أصبح مباشرًا وواضحًا.

تنفيذ المخطط في MySQL

تم إنشاء الجداول باستخدام أوامر:

CREATE TABLE

مع استخدام:

- Primary Keys
- Foreign Keys

للحفاظ على:

- سلامة العلاقات
- ترابط البيانات

إنشاء مخطط ERD

بعد إنشاء الجداول، تم استخدام ميزة:

Reverse Engineer

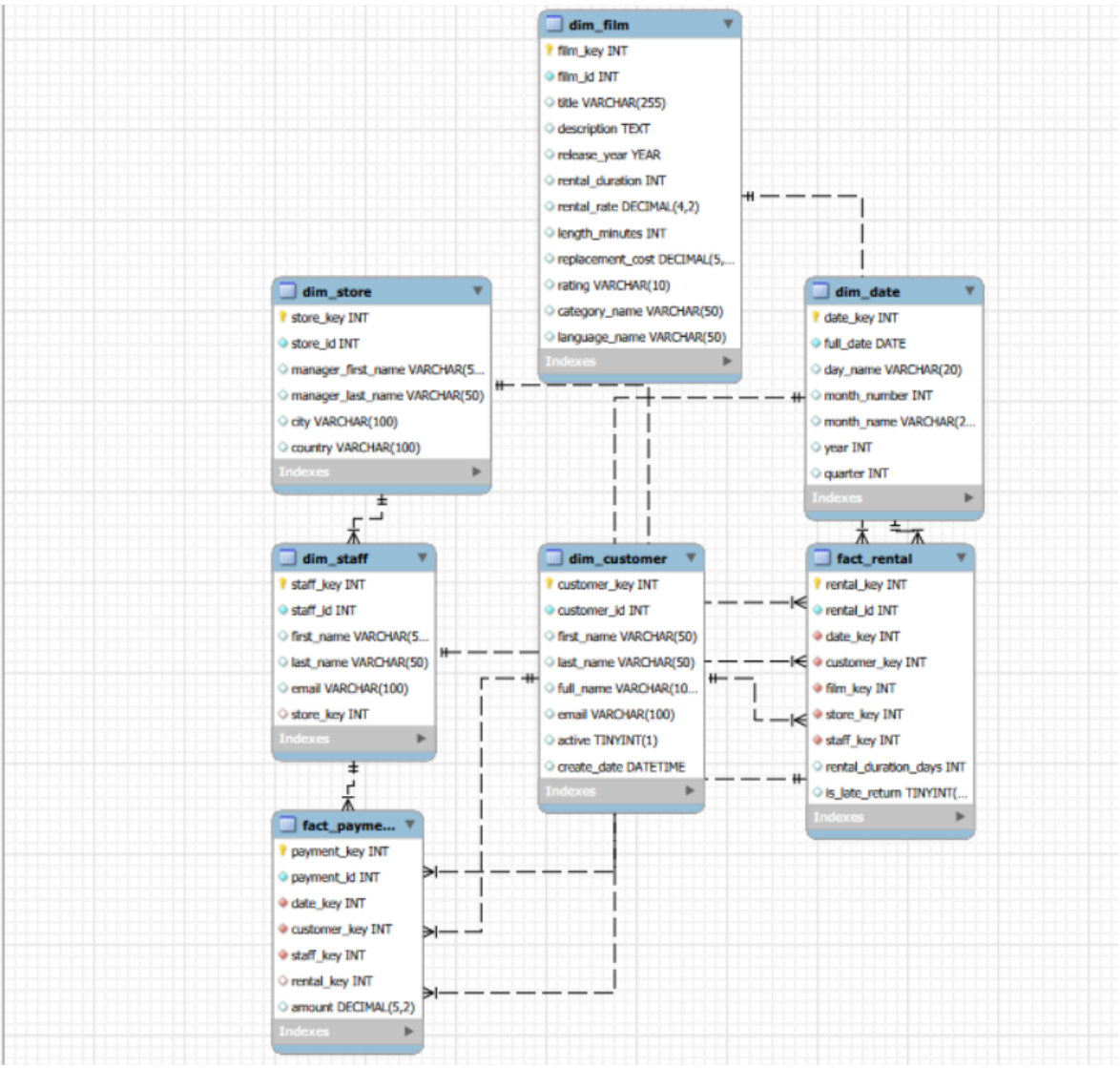
داخل MySQL Workbench لإنشاء مخطط ERD.

ماذا يُظهر الـ ERD؟

يوضح:

- العلاقات بين الجداول
- شكل الـ Star Schema
- ارتباط الـ Fact Tables بالـ Dimension Tables

ويظهر التصميم على شكل نجمة، لذلك يسمى Star Schema.



تصميم عملية ETL

تم تصميم عملية ETL لنقل البيانات من قاعدة الـ OLTP إلى الـ Data Warehouse.

ما معنى ETL؟

ETL تتكون من 3 مراحل:

1. **Extract ← استخراج البيانات**
2. **Transform ← تحويل البيانات**
3. **Load ← تحميل البيانات**

الأدوات المستخدمة

تم تنفيذ ETL باستخدام:

- **Python**
- **Pandas**
- **SQLAlchemy**

- MySQL

مرحلة Extract (استخراج البيانات)

في هذه المرحلة تم جلب البيانات من قاعدة البيانات الأصلية:

sakila

باستخدام:

- SQL Queries
- Pandas

الربط بين Python و MySQL

تم استخدام:

- SQLAlchemy
- PyMySQL

لإنشاء الاتصال بين بايثون وقاعدة البيانات.

الجداول التي تم استخراجها

تم استخراج البيانات من الجداول:

- customer
- rental
- payment
- film
- category
- film_category
- store
- staff
- language

طريقة استخراج البيانات

تم استخدام:

pd.read_sql()

مثال

```
customer_df = pd.read_sql(
    "SELECT * FROM customer",
```

engine

)

ماذا حدث بعد الاستخراج؟

تم تخزين البيانات داخل:

Pandas DataFrames

حتى يتم تعديلها وتحويلها لاحقًا.

مرحلة Transform (تحويل البيانات)

بعد استخراج البيانات، تم تنفيذ عدة عمليات تحويل حتى تصبح البيانات مناسبة للتحليل Datag Warehouse.

تحويلات العملاء (Customer Transformations)

إنشاء full_name

تم دمج:

- first_name
- last_name

في حقل جديد اسمه:

full_name

مثال

) = customer_df['full_name']

+ ' ' + customer_df['first_name']

customer_df['last_name']

)

تحويلات التأجير (Rental Transformations)

حساب مدة التأجير

تم حساب مدة التآجير باستخدام الفرق بين:

- return_date
- rental_date

مثال

```
) = rental_df['rental_duration_days']

- rental_df['return_date']

rental_df['rental_date']

dt.days.(
```

اكتشاف التأخير بالإرجاع

تم إنشاء حقل جديد:

is_late_return

لتحديد هل الفيلم أُعيد متأخرًا أم لا.

تحويلات الأفلام (Film Transformations)

دمج بيانات الفيلم والتصنيف

تم دمج بيانات:

- الأفلام
- التصنيفات

حتى تصبح التقارير أسهل وأسرع.

إعادة تسمية الأعمدة

بعض الحقول تم تغيير أسمائها لتناسب النموذج البُعدي.

أمثلة

length → length_minutes

name → category_name

تنظيف البيانات (Data Cleaning)

تم تنفيذ عمليات إضافية مثل:

- معالجة القيم المفقودة
- تحويل التواريخ إلى datetime
- ترتيب الأعمدة
- توحيد النصوص
- تجهيز الحقول التحليلية

لماذا هذه الخطوات مهمة؟

لأنها تجعل البيانات:

- منظمة
- نظيفة
- سهلة للتحليل

مرحلة Load (تحميل البيانات)

في هذه المرحلة تم إدخال البيانات المحولة إلى قاعدة بيانات الـ Data Warehouse:

movie_rental_dw

طريقة التحميل

تم استخدام:

()to_sql

تحميل جداول الأبعاد أولًا

تم تحميل:

- Dimension Tables

أولًا لأن:

- Fact Tables تعتمد عليها من خلال Foreign Keys.

الجداول التي تم تحميلها

- dim_customer
- dim_film

وغيرها.

مثال

)dim_customer.to_sql

,'dim_customer'

```
,dw_engine
,'if_exists='append
index=False
)
```

Surrogate Keys

تم استخدام:

AUTO_INCREMENT Surrogate Keys

بدل الاعتماد مباشرة على IDs الخاصة بالـ OLTP.

أمثلة

- customer_key
- film_key
- store_key
- staff_key

لماذا نستخدم Surrogate Keys؟

لأنها:

- تحسن الأداء
- تجعل الـ DW مستقلًا عن النظام الأصلي
- تسهّل إدارة العلاقات

تحميل جداول الحقائق

عند تحميل Fact Tables كان يجب:

- ربط IDs الأصلية
- مع الـ Surrogate Keys داخل الـ DW

وهذه العملية تسمى:

Key Mapping

الفائدة من ذلك

تضمن:

- سلامة العلاقات

- تناسق البيانات
- أداء أفضل

أدوات ETL المستخدمة

تم استخدام:

- Python
- Pandas
- SQLAlchemy
- PyMySQL
- MySQL Server
- VS Code / Jupyter Notebook

ملاحظة: ستجد اكواد البايثون والرسومات في ملف ال ETL

اعتبارات جودة البيانات (Data Quality Considerations)

جودة البيانات تعتبر جزءًا مهمًا جدًا من عملية ETL، لأن البيانات غير الدقيقة أو غير المتناسقة قد تؤثر على:

- التقارير التحليلية
- نتائج التحليل
- قرارات الإدارة

لذلك، أثناء عملية ETL تم تنفيذ عدة عمليات:

- فحص للبيانات
- تنظيف للبيانات
- التحقق من صحتها

قبل تحميلها إلى الـ Data Warehouse.

معالجة القيم المفقودة (Handling Missing Values)

بعض السجلات قد تحتوي على:

- قيم مفقودة
- أو NULL

خصوصًا في حقول مثل:

- return_date
- email
- وصف الفيلم

ماذا تم عمله؟

تم فحص القيم المفقودة أثناء مرحلة الـ Transform والتعامل معها حتى لا تؤثر على التحليل.

أمثلة

إذا كان return_date فارغًا

فهذا قد يعني:

أن الفيلم لم يتم إرجاعه بعد.

القيم النصية المفقودة

تم:

- تنظيفها
- أو استبدالها عند الحاجة

منع البيانات المكررة (Duplicate Data Prevention)

البيانات المكررة تسبب:

- أخطاء في التقارير
- نتائج تحليل غير صحيحة

ماذا تم عمله؟

أثناء ETL تم:

- فحص الصفوف المكررة
- منع تكرار السجلات قبل التحميل

كما تم استخدام:

- Primary Keys
- Foreign Keys

للحفاظ على تناسق البيانات.

دور الـ Surrogate Keys

ساعدت الـ Surrogate Keys على:

- تقليل الاعتماد على IDs الأصلية
- تجنب التكرار والمشاكل القادمة من OLTP

التحقق من أنواع البيانات (Data Type Validation)

تم التأكد أن كل عمود مخزن بالنوع الصحيح.

أمثلة

التواريخ

تم تحويلها إلى:

datetime

الحقول الرقمية

مثل:

payment amount

تم التأكد أنها أرقام فعلًا.

الحقول المنطقية

مثل:

is_late_return

تم التأكد أنها Boolean صحيحة.

الفائدة

هذا يساعد على:

- تحسين سرعة الاستعلامات
- زيادة دقة التحليل

توحيد النصوص (Standardizing Text Values)

تم توحيد بعض القيم النصية حتى تصبح التقارير متناسقة.

أمثلة

دمج أسماء العملاء

تم دمج:

- الاسم الأول
- الاسم الأخير

في حقل:

full_name

تنظيف أسماء التصنيفات واللغات

حتى تظهر بنفس التنسيق داخل التقارير.

توحيد أسماء المدن والدول

لمنع اختلاف الكتابة بين السجلات.

التحقق من الـ Foreign Keys

قبل تحميل جداول الحقائق، تم التأكد أن:

- كل Foreign Key موجود فعلًا داخل جدول البُعد المناسب
- كل Fact Table يشير إلى Surrogate Key صحيح

لماذا هذا مهم؟

حتى نحافظ على:

Referential Integrity

أي سلامة العلاقات بين الجداول.

التحقق من مدة التأجير (Rental Duration) (Validation)

تم التحقق من أن:

- مدة التأجير ليست سالبة
- حسابات التاريخ صحيحة
- حساب التأخير بالإرجاع دقيق

تناسق البيانات (Data Consistency)

تم أيضًا التأكد من وجود تناسق بين:

- بيانات الـ OLTP
- وبيانات الـ Data Warehouse

كيف؟

عن طريق:

- الحفاظ على العلاقات الصحيحة
- استخدام أسماء موحدة
- المحافظة على قواعد العمل أثناء التحويل

النتيجة

تطبيق هذه الفحوصات يساعد على:

- زيادة موثوقية التقارير
- تحسين دقة التحليل
- دعم الـ Business Intelligence بشكل صحيح

الخاتمة (Conclusion)

في هذا المشروع تم تصميم Data Warehouse بُعدي لنظام تأجير أفلام باستخدام قاعدة بيانات الـ OLTP الموجودة داخل:

sakila

ماذا يوضح المشروع؟

المشروع يوضح العملية الكاملة لتحويل:

- قاعدة بيانات تشغيلية OLTP
إلى:
- بيئة تحليلية مناسبة للتقارير والتحليل

تصميم الـ Star Schema

تم تصميم وتنفيذ:

Star Schema

باستخدام:

- SQL
- MySQL Workbench

ماذا احتوى المخطط؟

احتوى على:

- Fact Tables
- Dimension Tables

لدعم تحليل:

- التأجير
- الدفع
- العملاء
- الأفلام
- الفروع
- الموظفين

تنفيذ ETL

تم تنفيذ ETL باستخدام:

- Python
- Pandas
- SQLAlchemy

ماذا فعلت عملية ETL؟

نجحت في:

- استخراج البيانات من OLTP
- تحويل البيانات
- حساب القيم المشتقة

مثل:

- مدة التأجير
- التأخير بالإرجاع

ثم تحميل البيانات داخل الـ Data Warehouse.

التحليلات والرسومات

المشروع احتوى أيضًا على:

- رسومات تحليلية
- أمثلة تقارير

توضح كيف يساعد الـ Data Warehouse في:

- التحليل
- دعم القرار

الفرق بين OLTP و Data Warehouse

أنظمة OLTP

مخصصة لـ:

- العمليات اليومية
- المعاملات السريعة

مثل:

- التأجير
- الدفع
- التحديثات المستمرة

أنظمة Data Warehouse

مخصصة لـ:

- التحليل
- التقارير
- دراسة البيانات التاريخية

الخلاصة النهائية

الـ Data Warehouse المقترح يوفر:

- هيكل بسيط
- أداء أفضل
- تحليل أسهل

لعمليات تأجير الأفلام، كما يدعم مستقبلًا:

- Business Intelligence
- Data Analytics
- التقارير المتقدمة
- اتخاذ القرار بشكل أفضل.