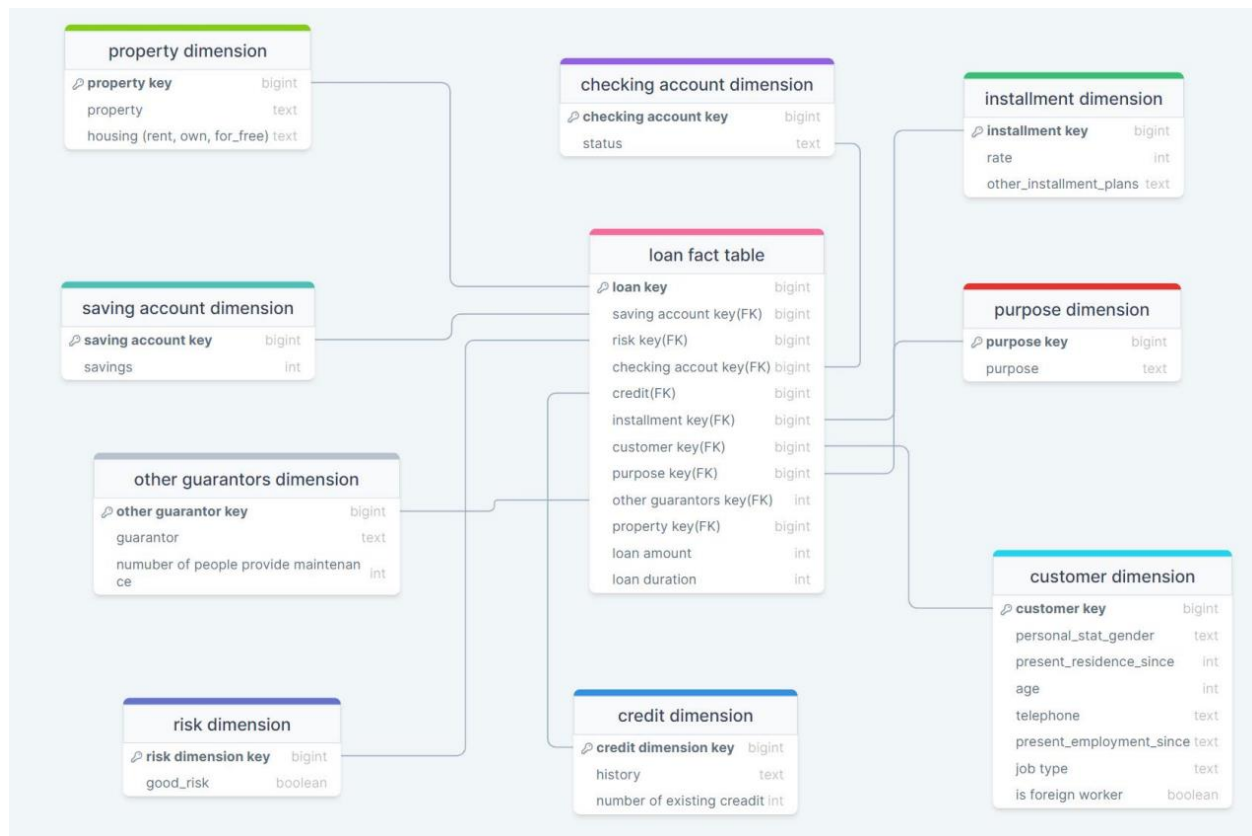


schematic staging plan



All the data from the same data source: German Credit Scoring Data

<https://www.kaggle.com/datasets/elsnkazm/german-credit-scoring-data>

(discussed with TA, single abundant data sources is acceptable)

the source data is well organized (binning, label) already, but I just carried out data cleaning procedure for the raw data. From the data source description, I grabbed the valid values/range for each column and applied the filters (made by valid values) to them, for example:

```
# remove out range value other_installment_plans
other_installment_plans_values=['none', 'bank', 'store']
df=df[df['other_installment_plans'].isin(other_installment_plans_values)]
```

And delete missing data row (if any). I deleted them because there were enough data:

Check for null value

```
df=df[~df.isnull().any(axis=1)]
```

348] ✓ 0.0s

I also separate some columns to better organize the data in data mart:

```
df[['gender', 'marriage']] = df['personal_stat_gender'].str.split(':', expand=True)
df=df.drop('personal_stat_gender', axis=1)
```

9] ✓ 0.0s

After creating and relating all the dimensions and fact table, I created database instance and load the data into it:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane shows the database structure, including a 'public' schema with a 'loan' table. The 'Data Output' pane at the bottom displays the data loaded from the CSV file, showing columns like 'id', 'saving_account_id', 'checking_account_id', 'credit_id', 'installment_id', 'customer_id', 'purpose_id', 'other_guarantors_id', 'property_id', 'duration', 'loan_amt', and 'good_risk'. The data is organized into rows, with the first row being the header and subsequent rows containing numerical values and a boolean 'good_risk' value.

```
CREATE TABLE loan (
  id INT PRIMARY KEY NOT NULL,
  saving_account_id INT REFERENCES saving_account(id) NOT NULL,
  checking_account_id INT REFERENCES checking_account(id) NOT NULL,
  credit_id INT REFERENCES credit(id) NOT NULL,
  installment_id INT REFERENCES installment(id) NOT NULL,
  customer_id INT REFERENCES customer(id) NOT NULL,
  purpose_id INT REFERENCES purpose(id) NOT NULL,
  other_guarantors_id INT REFERENCES other_guarantors(id) NOT NULL,
  property_id INT REFERENCES property(id) NOT NULL,
  duration INT NOT NULL,
  loan_amt INT NOT NULL,
  good_risk BOOL NOT NULL
);
```

copy loan from 'C:\Users\chent\OneDrive\Desktop\CS14142\Project\DataMart\loan_fact_table.csv' delimiter ','

SELECT * FROM loan;

id	saving_account_id	checking_account_id	credit_id	installment_id	customer_id	purpose_id	other_guarantors_id	property_id	duration	loan_amt	good_risk
127	127	2	1	127	127	1	127	127	12	701	true
128	128	2	2	128	128	8	128	128	12	639	false
129	129	2	2	129	129	5	129	129	12	1860	true
130	130	2	1	130	130	4	130	130	12	3499	false
131	131	1	2	131	131	4	131	131	48	8487	true
132	132	2	1	132	132	2	132	132	36	6887	false
133	133	2	3	133	133	3	133	133	15	2708	true
134	134	2	3	134	134	3	134	134	18	1984	true
135	135	5	3	135	135	1	135	135	60	10144	true
136	136	1	3	136	136	1	136	136	12	1240	true
137	137	4	3	137	137	5	137	137	27	8613	true
138	138	3	2	138	138	1	138	138	12	766	false
139	139	1	2	139	139	1	139	139	15	2728	true

Total rows: 1000 of 1000 Query complete 00:00:00.195 Ln 13, Col 25

