

# REPORT PROGETTO

## 1. DATASET ANALYSIS

Per questa prima parte del progetto abbiamo individuato i dataset per la nostra ricerca.

Per affrontare il tema, il nostro team si è voluto focalizzare sull'analisi del gender gap nel mondo accademico partendo dalle materie ICT, monitorando l'andamento nella crescita lavorativa e arrivando a concentrarci su specifici indici di riferimento quali:

- Disuguaglianza salariale di genere (%)
- Donne in posizione manageriale di vertice (%)
- Donne nel CDA
- N. Donne dipendenti
- N. Uomini dipendenti

I dataset scelti hanno fatto riferimento quindi a:

- 1) DNF: (Dataset minimo da utilizzare, fornитoci tra le risorse del progetto -  
<https://www.osservatoriодnf.it/production/dashboard/dashboard2024.php?F=fase3&VAR=T&TYPEVAR=TOTAL>)
- 2) Eurostat: (<https://ec.europa.eu/eurostat/web/sdi/database/gender%20equality>)
- 3) Dati MUR: (<https://dati-ustat.mur.gov.it/dataset/dati-per-bilancio-di-genere>)

Antecedente alla scelta dei dataset c'è stata una fase di documentazione sul fenomeno, prendendo come riferimento il pdf che illustrava l'organizzazione del progetto.

Per effettuare la fase di analisi ed estrazione del materiale richiesto ci siamo avvalsi di script in python, progettati ad hoc e di un software di visualizzazione dati chiamato "Knime", al fine di estrarre il numero di attributi per colonna ed organizzarli in modo ascendente da quello con più valori NULL a quelli con meno.

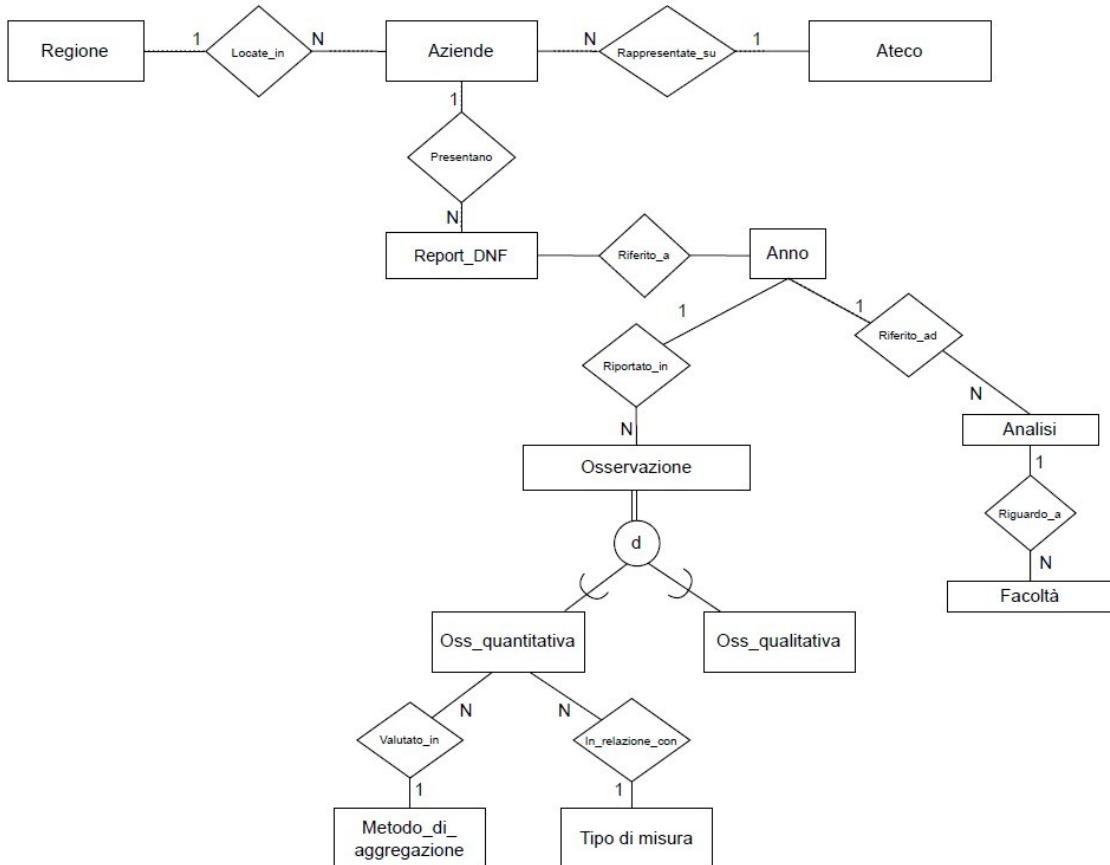
In seguito, dopo aver chiarito la struttura completa delle tabelle e l'organizzazione del database, ci siamo avvalsi di schemi concettuali ER.

Per la gestione del progetto è stata utilizzata la piattaforma GitHub, che ha permesso la condivisione del codice sorgente, degli script e dei dataset in formato CSV.

<https://github.com/nicolocarcagni/genderhack2025>

## 2. Database Design and Implementation

Per la seconda fase del progetto abbiamo rappresentato i modelli ER (inseriti sulla piattaforma Github) ed infine il modello EER complessivo.



Continuando, abbiamo definito per ogni entità i rispettivi attributi:

REGIONE (id\_regione, nome)

AZIENDE (id\_azienda, nome, cod\_ateco, cod\_region)

ATECO (id\_ateco, settore)

REPORT\_DNF (id\_report\_dnf, valore, cod\_azienda, cod\_dnf, cod\_anno)

DNF(id\_dnf, nome)

ANALISI (id\_analisi, num\_iscritti\_m, num\_iscritti\_f, cod\_facolta, cod\_anno)

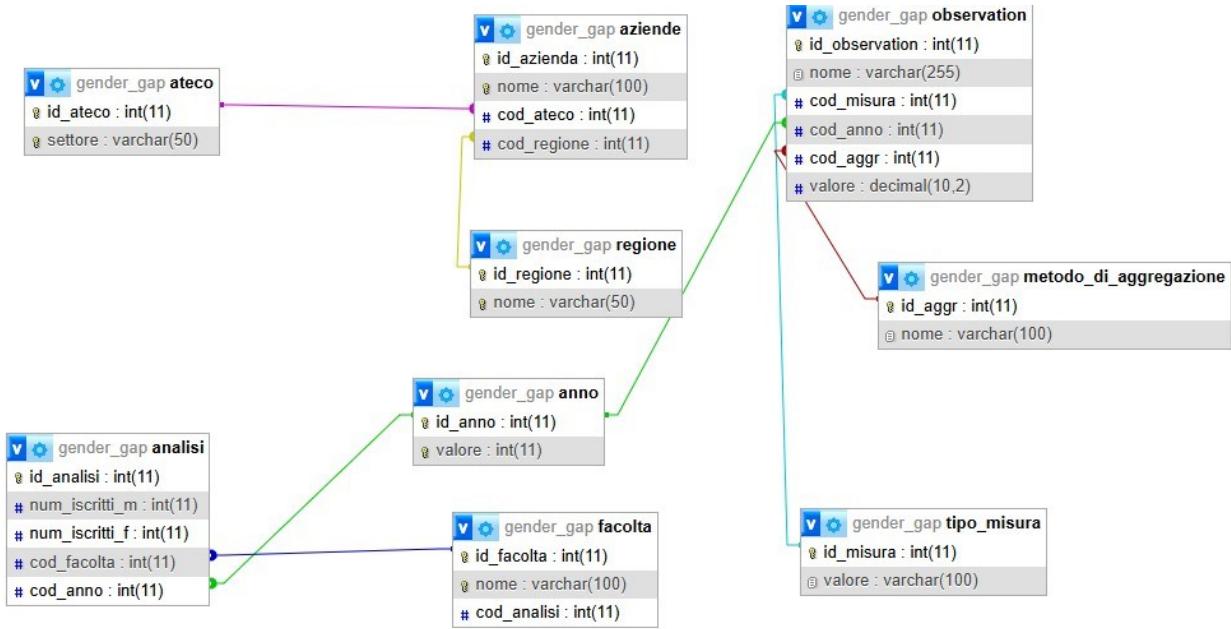
FACOLTA (id\_facolta, nome, cod\_analisi)

METODO\_DI\_AGGREGAZIONE (id\_aggr, nome)

OBSERVATION (id\_observation, nome, cod\_misura, cod\_anno, cod\_aggr, valore)

REGIONE (id\_regione, nome)

TIPO\_MISURA (id\_misura, valore)



Rappresentazione parziale dello schema fisico del DB

Seguendo questo schema è facile arrivare alla conclusione che il database è in forma normale, ovvero:

ogni colonna contiene valori atomici e non ci sono chiavi composte per cui, automaticamente, troviamo corrispondenza nella seconda e nella terza forma normale.

(I commenti riguardo l'analisi del modello e dei dataset sono stati inseriti sulla piattaforma Github)

### 3. Data Pipeline Development

Il terzo punto per l'analisi dei dati richiedeva la creazione e il corretto inserimento dei dati estratti nel database.

Per la corretta estrazione e formattazione, ci siamo avvalse di script python, generati per lo scopo, che ci hanno portato a formattare correttamente i valori e la struttura dei dataset.

#### QUERY SQL PER LA CREAZIONE DELLE TABELLE (sintassi generale utilizzata):

CREATE TABLE (

Attributo tipo\_attributo (nel nostro caso: INT, VARCHAR, DECIMAL) NOT NULL (indicatore utile a specificare l'unicivocità del campo e rientrare nella prima forma normale) PRIMARY KEY

);

Nella creazione della tabella è importante specificare quali campi verranno utilizzati come FOREIGN KEY nella sintassi operativa.

CONSTRAINT fk\_nome

foreign key (attributo) REFERENCES tabella (attributo di riferimento della tabella)

#### **QUERY SQL PER IL RIEMPIMENTO DELLE TABELLE (sintassi generale utilizzata):**

LOAD DATA INFILE 'percorso'

INTO TABLE `table`

FIELDS TERMINATED BY ','

ENCLOSED BY ""

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

#### **4. ESTRAZIONE DATI**

Infine, come ultima parte del progetto, abbiamo applicato delle query sql per interrogare il database creato per fornirci tutti i dati necessari per stilare il report tecnico sul fenomeno:

#### **QUERY PER OTTENERE IL GAP SALARIALE DI GENERE**

SELECT

d.nome AS indice,

rd.valore AS valore\_di\_riferimento,

a.nome\_azienda,

an.valore AS anno

FROM

report\_dnf AS rd

JOIN

dnf AS d ON rd.cod\_dnf = d.id\_dnf

JOIN

azienda AS a ON rd.cod\_azienda = a.id\_azienda

JOIN

anno AS an ON rd.cod\_anno = an.id\_anno

WHERE  
d.nome LIKE '%Disuguaglianza\_salariale%'  
AND a.ateco IN ('5', '6', '14', '20')

ORDER BY  
rd.valore DESC;

#### **QUERY PER OTTENERE LA PERCENTUALE DI DONNE IN POSIZIONE MANAGERIALE DI VERTICE**

SELECT  
a.nome,  
rd.valore  
FROM  
report\_dnf AS rd  
JOIN  
dnf AS d ON rd.cod\_dnf = d.id\_dnf  
JOIN  
aziende AS a ON rd.cod\_azienda = a.id\_azienda  
JOIN  
anno AS an ON rd.cod\_anno = an.id\_anno  
WHERE  
d.nome LIKE '%Donne%manageriali%vertice%'  
AND a.cod\_ateco IN ('5', '6', '14', '20')  
ORDER BY  
rd.valore DESC;

## **QUERY PER OTTENERE LA PERCENTUALE DI DONNE NEL CDA**

```
SELECT  
    a.nome,  
    rd.valore  
FROM  
    report_dnf AS rd  
JOIN  
    dnf AS d ON rd.cod_dnf = d.id_dnf  
JOIN  
    aziende AS a ON rd.cod_azienda = a.id_azienda  
JOIN  
    anno AS an ON rd.cod_anno = an.id_anno  
WHERE  
    d.nome LIKE '%Donne%nel%CDA'  
  
AND a.cod_ateco IN ('5', '6', '14', '20')  
ORDER BY  
    rd.valore DESC;
```

## **QUERY PER OTTENERE IL NUMERO DI DONNE DIPENDENTI**

```
SELECT  
    a.cod_ateco,  
    rd.valore  
FROM  
    report_dnf AS rd  
JOIN  
    dnf AS d ON rd.cod_dnf = d.id_dnf  
JOIN  
    aziende AS a ON rd.cod_azienda = a.id_azienda
```

JOIN  
anno AS an ON rd.cod\_anno = an.id\_anno

WHERE  
d.nome LIKE '%N%DONNE%DIPENDENTI%'

AND a.cod\_ateco IN ('5', '6', '14', '20')

ORDER BY  
rd.valore DESC;

#### **QUERY PER OTTENERE IL NUMERO DI UOMINI DIPENDENTI**

SELECT  
a.cod\_ateco,  
rd.valore  
FROM  
report\_dnf AS rd  
JOIN  
dnf AS d ON rd.cod\_dnf = d.id\_dnf  
JOIN  
aziende AS a ON rd.cod\_azienda = a.id\_azienda  
JOIN  
anno AS an ON rd.cod\_anno = an.id\_anno  
WHERE  
d.nome LIKE '%N%UOMINI%DIPENDENTI%'

AND a.cod\_ateco IN ('5', '6', '14', '20')

ORDER BY  
rd.valore DESC;

#### **QUERY PER OTTENERE IL NUMERO DI FEMMINE ISCRITTE ALLE FACOLTÀ STEM**

```
SELECT  
    f.nome AS facoltà,  
    a.num_iscritti_f AS NUMERO_ISCRITTE  
  
FROM  
    analisi as a  
  
JOIN  
    facolta as f ON a.cod_facolta = f.id_facolta  
  
JOIN  
    anno AS an ON a.cod_anno = an.id_anno  
  
WHERE  
    an.valore = 2013;
```

#### **QUERY PER OTTENERE IL NUMERO DI UOMINI ISCRITTI ALLE FACOLTÀ STEM**

```
SELECT  
    f.nome AS facoltà,  
    a.num_iscritti_m AS NUMERO_ISCRITTI  
  
FROM  
    analisi as a  
  
JOIN  
    facolta as f ON a.cod_facolta = f.id_facolta  
  
JOIN  
    anno AS an ON a.cod_anno = an.id_anno  
  
WHERE  
    an.valore = 2013;
```

## QUERY PER OTTENERE LA SOLUZIONE MACROECONOMICA

SELECT

a.valore AS Anno,

CASE

WHEN o.cod\_misura = 1 THEN 'Differenza nel tasso di Occupazione (%)'

WHEN o.cod\_misura = 2 THEN 'Differenza nel Part-Time (%)'

WHEN o.cod\_misura = 3 THEN 'Differenza nel Tempo Determinato (%)'

WHEN o.cod\_misura = 4 THEN 'Differenza nel tasso di Disoccupazione (%)'

ELSE 'Altro Indicatore'

END AS Codice\_Misura,

o.valore AS Valore\_Punti\_Percentuali

FROM

observation AS o

JOIN

anno AS a ON o.cod\_anno = a.id\_anno --

WHERE

o.nome LIKE 'IT\_%'

AND o.cod\_misura IN (1, 2, 3, 4)

ORDER BY

Anno ASC;