**DATASETS:**

1. **Sports and Recreation Clubs**
2. **Multi-Use Community Centres**
3. **Accessible Parking Spaces**

**PREFIXES:**

PREFIX csv: <http://www.semanticweb.org/KDE#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX math:<http://www.w3.org/2005/xpath-functions/math#>

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX geo: <http://www.opengis.net/ont/geosparql#>

PREFIX cs: <http://purl.org/vocab/changeset/schema#>

1. **What are SDCC (South Dublin City Council) owned Sports and Recreation Clubs &**

**Multi-Use Community Centres?**

SELECT ?Name\_Of\_Center

WHERE {

?subject csv:isSdccOwned "Yes".

?subject csv:hasName ?Name\_Of\_Center.

}

1. **Give names of Sports and Recreation Clubs & Multi-Use Community Centres present**

**In DUBLIN 12**

SELECT ?Name\_Of\_Center ?Address

WHERE {

?subject csv:hasAddress ?Address.

?center csv:hasContactInfo ?subject.

?center csv:hasName ?Name\_Of\_Center.

FILTER regex(?Address, "DUBLIN 12", "i")

}

1. **Which Sports and Recreation Clubs & Multi-Use Community Centres have meeting rooms to use?**

SELECT ?Name\_Of\_Center ?Is\_Present

WHERE {

?subject csv:hasMeetingRooms ?Is\_Present.

?center csv:hasFacilities ?subject.

?center csv:hasName ?Name\_Of\_Center.

FILTER(?Is\_Present = "Yes")

}

1. **What are the counts of Sports and Recreation Clubs & Multi-Use Community centres were created by Community and ESRI (The Economic and Social Research Institute)?**

SELECT ?Creator (xsd:string(COUNT(?name)) AS ?Count)

WHERE {

?subject csv:createdBy ?Creator.

?subject csv:hasName ?name.

}GROUP BY ?Creator

1. **Which Sports and Recreation Clubs & Multi-Use Community Centres have**

**coffee docks?**

SELECT ?Name\_Of\_Center ?Is\_Present

WHERE {

?subject csv:hasCoffeeDock ?Is\_Present.

?center csv:hasFacilities ?subject.

?center csv:hasName ?Name\_Of\_Center

FILTER(?Is\_Present = "Yes")

}

1. **What are the parking areas available near a particular Sports and Recreation Club within distance of x meters ?**

SELECT ?Name\_Of\_Center ?Distance\_In\_Meters

WHERE {

?subject csv:hasX ?x.

?subject csv:hasY ?y.

?parkingrecord csv:hasCoordinates ?subject.

?parkingrecord rdf:type csv:ParkingSpace.

?parkingrecord csv:hasLocationName ?Name\_Of\_Center.

?parkingrecord csv:hasSpaceType ?type.

{

SELECT ?selectedx ?selectedy

WHERE {

?sub csv:hasX ?selectedx.

?sub csv:hasY ?selectedy.

?center csv:hasCoordinates ?sub.

?center csv:hasName "MARK'S CELTIC FOOTBALL CLUB".

}

}

BIND ((xsd:decimal(?selectedx) - xsd:decimal(?x)) \* 0.0174533 AS ?phi)

BIND ((xsd:decimal(?selectedy) - xsd:decimal(?y)) \* 0.0174533 AS ?lambda)

BIND ((xsd:decimal(?x)) \* 0.0174533 AS ?lat1radians)

BIND ((xsd:decimal(?selectedx)) \* 0.0174533 AS ?lat2radians)

BIND(math:sin(?phi / xsd:decimal(2)) \* math:sin(?phi / xsd:decimal(2)) + math:cos(?lat1radians) \*

math:cos(?lat2radians) \* math:sin(?lambda / xsd:decimal(2)) \* math:sin(?lambda / xsd:decimal(2))

AS ?a)

BIND(xsd:decimal(2) \* math:atan2(math:sqrt(?a),math:sqrt(1-?a)) AS ?c)

BIND(xsd:decimal(6371000) \* xsd:decimal(?c) AS ?distance)

BIND(xsd:string(?distance) AS ?Distance\_In\_Meters)

FILTER(?distance < xsd:decimal(4000))

} ORDER BY ?distance

1. **What are the different kinds of Space-type available for parking near a particular Sports and Recreation Club and what are the counts of each of them?**

SELECT ?Type (xsd:string(Count(?Type)) AS ?Count)

WHERE {

?subject csv:hasX ?x.

?subject csv:hasY ?y.

?parkingrecord csv:hasCoordinates ?subject.

?parkingrecord rdf:type csv:ParkingSpace.

?parkingrecord csv:hasLocationName ?name.

?parkingrecord csv:hasSpaceType ?Type.

{

SELECT ?selectedx ?selectedy

WHERE {

?sub csv:hasX ?selectedx.

?sub csv:hasY ?selectedy.

?center csv:hasCoordinates ?sub.

?center csv:hasName "NEILSTOWN COMMUNITY CENTRE".

}

}

# To get the distance between two coordinates in metres

# ref: https://www.movable-type.co.uk/scripts/latlong.html

BIND ((xsd:decimal(?selectedx) - xsd:decimal(?x)) \* 0.0174533 AS ?phi)

BIND ((xsd:decimal(?selectedy) - xsd:decimal(?y)) \* 0.0174533 AS ?lambda)

BIND ((xsd:decimal(?x)) \* 0.0174533 AS ?lat1radians)

BIND ((xsd:decimal(?selectedx)) \* 0.0174533 AS ?lat2radians)

BIND(math:sin(?phi / xsd:decimal(2)) \* math:sin(?phi / xsd:decimal(2)) + math:cos(?lat1radians) \*

math:cos(?lat2radians) \* math:sin(?lambda / xsd:decimal(2)) \* math:sin(?lambda / xsd:decimal(2))

AS ?a)

BIND(xsd:decimal(2) \* math:atan2(math:sqrt(?a),math:sqrt(1-?a)) AS ?c)

BIND(xsd:decimal(6371000) \* xsd:decimal(?c) AS ?distance)

BIND(xsd:string(?distance) AS ?dist)

FILTER(?distance < xsd:decimal(2000))

}GROUPBY ?Type

1. **What are the Sports and Recreation Clubs & Multi-Use Community Centres**

**that have missing website information?**

SELECT ?name ?subject

WHERE {

?record csv:hasOnlinePresence ?subject.

?record csv:hasName ?name.

FILTER(NOT EXISTS { ?subject csv:hasWebsite ?site })

}

1. **What are disability accessible Sports and Recreation Clubs & Multi-Use**

**Community Centres in Dublin 16?**

SELECT ?Name\_Of\_Center ?Address

WHERE {

?subject csv:hasAddress ?Address.

?center csv:hasContactInfo ?subject.

?center csv:hasName ?Name\_Of\_Center.

FILTER regex(?Address, "DUBLIN 12", "i")

}

1. **What are the details of nearest parking area available near a particular Sports and**

**Recreation Club and Give the number of available parking spaces?**

SELECT ?Name ?Distance (xsd:string(?numberofspaces) AS ?Number\_Of\_Spaces) ?Type\_Of\_Space

WHERE {

?subject csv:hasX ?x.

?subject csv:hasY ?y.

?parkingrecord csv:hasCoordinates ?subject.

?parkingrecord rdf:type csv:ParkingSpace.

?parkingrecord csv:hasLocationName ?Name.

?parkingrecord csv:hasSpaceType ?Type\_Of\_Space.

?parkingrecord csv:numberOfSpaces ?numberofspaces

{

SELECT ?selectedx ?selectedy

WHERE {

?sub csv:hasX ?selectedx.

?sub csv:hasY ?selectedy.

?center csv:hasCoordinates ?sub.

?center csv:hasName "SAINT MARY'S RUGBY FOOTBALL CLUB".

}

}

# To get the distance between two coordinates in metres

# ref: https://www.movable-type.co.uk/scripts/latlong.html

BIND ((xsd:decimal(?selectedx) - xsd:decimal(?x)) \* 0.0174533 AS ?phi)

BIND ((xsd:decimal(?selectedy) - xsd:decimal(?y)) \* 0.0174533 AS ?lambda)

BIND ((xsd:decimal(?x)) \* 0.0174533 AS ?lat1radians)

BIND ((xsd:decimal(?selectedx)) \* 0.0174533 AS ?lat2radians)

BIND(math:sin(?phi / xsd:decimal(2)) \* math:sin(?phi / xsd:decimal(2)) + math:cos(?lat1radians) \*

math:cos(?lat2radians) \* math:sin(?lambda / xsd:decimal(2)) \* math:sin(?lambda / xsd:decimal(2))

AS ?a)

BIND(xsd:decimal(2) \* math:atan2(math:sqrt(?a),math:sqrt(1-?a)) AS ?c)

BIND(xsd:decimal(6371000) \* xsd:decimal(?c) AS ?distance)

BIND(xsd:string(?distance) AS ?Distance)

FILTER(?distance < xsd:decimal(2000))

}ORDER BY ?distance LIMIT 1 OFFSET 1